



**TUGAS AKHIR - TE 141599**

**PENGOLAHAN CITRA UNTUK MENGUKUR DIAMETER  
TERKECIL KAYU GUNA MENGATASI RUGI AKIBAT  
KESALAHAN PENGUKURAN PADA INDUSTRI KAYU**

**RIFYAL RACHMAT  
NRP 2211 100 082**

**Dosen Pembimbing  
Ronny Mardiyanto, ST., MT., Ph.D.  
Fajar Budiman, ST., M.Sc.**

**JURUSAN TEKNIK ELEKTRO  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember  
Surabaya 2015**

*Halaman ini sengaja dikosongkan*



**FINAL PROJECT - TE 141599**

**IMAGE PROCESSING FOR MEASURING WOOD'S SMALLEST  
DIAMETER TO OVERCOME LOSS DUE TO ERRORS IN THE  
MEASUREMENT OF WOOD INDUSTRY**

RIFYAL RACHMAT  
NRP 2211 100 082

Supervisor  
Ronny Mardiyanto, ST., MT., Ph.D.  
Fajar Budiman, ST., M.Sc.

DEPARTMENT OF ELECTRICAL ENGINEERING  
Faculty of Industrial Technology  
Sepuluh Nopember Institute of Technology  
Surabaya 2015

*Halaman ini sengaja dikosongkan*

**PENGOLAHAN CITRA UNTUK MENGUKUR DIAMETER  
TERKECIL KAYU GUNA MENGATASI RUGI AKIBAT  
KESALAHAN PENGUKURAN PADA INDUSTRI KAYU**

**TUGAS AKHIR**

**Diajukan Guna Memenuhi Sebagian Persyaratan Untuk  
Memperoleh Gelar Sarjana Teknik  
Pada  
Bidang Studi Elektronika  
Jurusan Teknik Elektro  
Institut Teknologi Sepuluh Nopember**

**Menyetujui**

**Dosen Pembimbing I,**

**Dosen Pembimbing II,**

**Ronny Mardiyanto, ST., MT., Ph.D.  
NIP: 198101182603121003**

**Fajar Budiman, ST., M.Sc.  
NIP: 1986070720140411001**

**SURABAYA  
JULI, 2015**

**JURUSAN  
TEKNIK ELEKTRO**

## ABSTRAK

### **Pengolahan Citra untuk Mengukur Diameter Terkecil Kayu Guna Mengatasi Rugi Akibat Kesalahan Pengukuran pada Industri Kayu**

**Nama** : Rifyal Rachmat  
**NRP** : 2211 100 082  
**Dosen Pembimbing 1** : Ronny Mardiyanto, ST.,MT.,Ph.D.  
**Dosen Pembimbing 2** : Fajar Budiman, ST.,M.Sc.

Kayu merupakan hasil hutan sebagai bahan dasar dari produk jadi seperti lemari, kursi, dan produk olahan lain. Pada industri pengolahan kayu, metode penentuan harga umumnya didasarkan terhadap perhitungan isi kayu (volume) dimana nilai tersebut diawali dengan perhitungan diameter terkecil. Pekerja lapangan pada usaha kayu menggunakan nilai dari diameter terkecil untuk dijadikan acuan perhitungan dengan penggaris. Akan tetapi pengukuran ini kurang akurat karena kayu gelondong mempunyai bentuk yang tidak bulat sempurna.

Padahal, selisih satu centimeter saja dapat menentukan grade kayu dan berpengaruh terhadap harga kayu gelondong tersebut. Penelitian menggunakan algoritma pengolahan citra telah dilakukan maupun dengan spesifikasi tentang permukaan kayu. Teknologi yang dirancang memanfaatkan penginderaan mesin visual yang dilakukan oleh kamera untuk melakukan penghitungan otomatis terhadap nilai kayu dengan program OpenCV.

Pada tugas akhir ini, sebuah alat tepat guna dengan raspberry pi yang akan menjadi standarisasi pengukuran pada usaha kayu dengan menggunakan pengolahan citra digital. Devais yang dibuat sangat portabel dan berhasil mengukur diameter terkecil kayu dengan akurasi pengukuran 97% menggunakan metode kontur dan *bounding box*. Metode tersebut lebih baik dari metode lingkaran Hough dengan akurasi 89%. Perbaikan sistem dapat dilakukan pada deteksi obyek melalui pencahayaan dan segmentasi kontur.

**Kata kunci:** Pengukuran kayu, pengolahan citra, diameter terkecil kayu, deteksi kontur, *bounding box*

*Halaman ini sengaja dikosongkan*

## ABSTRACT

### Image Processing for Measuring Wood's Smallest Diameter to Overcome Loss Due to Errors in The Measurement of Wood Industry

Name : Rifyal Rachmat  
NRP : 2211 100 082  
Supervisor 1 : Ronny Mardiyanto, ST., MT., Ph.D.  
Supervisor 2 : Fajar Budiman, ST., M.Sc.

Wood or timber is a forest products as a basic ingredient of furniture such as cabinets, chairs, and other refined products. It is widely used by all people as households, offices, and shops. However, there are problem faced by the wood business activity, namely pricing of round wood or log. Traditional pricing method, generally based on the calculation of timber content (volume) where the value begins with the diameter calculation

Potential errors when measuring the diameter of a round wooden occurs because that does not always form a perfect circle. In fact, the difference of one centimeter alone can determine the grade timber and logs an effect on the price and potentially cause the loss of the industry. Researches based on image processing algorithm have been done or with the speziallitation of wood surface. Those technology will be build with image processing from the camera to automatic calculate for diameter's value combined with OpenCV algorithm.

In this final project, an appropriate device with *raspberry pi* that will be standardizing the measurement of the timber by using *image processing* method. Device is portable and this can measure the smallest diameter of the wood with 97% accuration using contours and bounding box method. It is better than hough circle method with 89% accuration. System upgrade potentially do at object detection with lighting and contours segmentation.

**Keyword:** *Measurement of wood, image processing, wood's smallest diameter, contour detection, bounding box*



*Halaman ini sengaja dikosongkan*

## KATA PENGANTAR

Teriring salam dan do'a sehingga Allah SWT senantiasa melimpahkan rahmat dan hidayah-Nya atas segala nikmat dan berkah, yang tak terkira kepada penulis, sehingga penulis mampu menyelesaikan Tugas Akhir dengan judul :

### **Pengolahan Citra untuk Mengukur Diameter Terkecil Kayu Guna Mengatasi Rugi Akibat Kesalahan Pengukuran pada Industri Kayu**

Tugas Akhir ini adalah sebagai salah satu persyaratan untuk memperoleh gelar sarjana teknik pada jurusan Teknik Elektro Institut Teknologi Sepuluh Nopember dengan Bidang Studi Elektronika

Oleh karena itu, pada kesempatan ini penulis menyampaikan terima kasih dan penghargaan setinggi tingginya kepada:

1. Bapak Ronny Mardiyanto dan Bapak Fajar Budiman selaku dosen pembimbing Tugas Akhir yang telah memberi bimbingan, kejelasan, nasehat, dan kemudahan dalam penyelesaian Tugas Akhir ini.
2. Bapak Tri Arief Sardjono selaku Ketua Jurusan dan dosen penulis.
3. Ibu Enny Rachmawati, dan Bapak Nurul Huda yang telah memberi dukungan moril, nasehat, semangat, doa, dan materil.
4. Mbak Aliffia Permata dan keluarga yang ikut memberi dukungan moril dan doa untuk kelancaran studi penulis.
5. Keluarga besar laboratorium B202 yang telah ikut memberi semangat, canda tawa, dan saling berbagi ilmu.
6. Keluarga besar Himpunan Mahasiswa Islam Sepuluh Nopember, yang telah menjadi teman dan sahabat penulis.
7. Keluarga besar ITS EXPO, yang telah membagi semangat dan pengalaman.
8. PT. Indofuritama Raya yang telah memberikan informasi dan berkeluh kesah tentang masalah di perusahaannya sehingga memberikan inspirasi ide.
9. Tim *MTC Electrical Art* yang membangun budaya belajar elektronika.
10. Teman-teman angkatan Elektro 2011 ELVN yang telah menyumbangkan cerita kepada kehidupan penulis.
11. Serta semua pihak yang telah membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis berharap para pembaca Tugas Akhir ini bersedia memberikan kritik, saran, dan masukan agar selanjutnya menambah manfaat Tugas Akhir dan menjadi sumbangan ilmu pengetahuan bidang elektronika.

Semoga laporan Tugas Akhir ini dapat bermanfaat dan bisa dikembangkan.

Surabaya, Juli 2015

Penulis

## DAFTAR ISI

HALAMAN JUDUL	
LEMBAR PERNYATAAN KEASLIAN	
LEMBAR PENGESAHAN	
ABSTRAK .....	i
ABSTRACT (Versi bahasa Inggris) .....	iii
KATA PENGANTAR .....	v
DAFTAR ISI .....	vii
DAFTAR GAMBAR .....	ix
DAFTAR TABEL .....	xiii
BAB 1 PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Perumusan Masalah .....	2
1.3 Tujuan Penelitian .....	2
1.4 Batasan Masalah .....	2
1.5 Metodologi Penelitian .....	3
1.6 Sistematika Penulisan .....	4
1.7 Relevansi .....	4
BAB 2 TINJAUAN PUSTAKA .....	5
2.1 Perhitungan Diameter Kayu Secara Konvensional .....	5
2.2 Penginderaan Visual .....	7
2.3 Interaksi Manusia- Mesin .....	7
2.4 Perangkat Raspberry Pi sebagai Single Board Circuit .....	7
2.4.1 Memulai Perangkat Raspberry Pi 2 .....	9
2.5 Raspbian .....	10
2.6 Install OpenCV pada Raspberry Pi 2 .....	12
2.7 Menggunakan Code::Blocks sebagai Compiler .....	15
2.8 Raspberry Pi Camera (RaspiCam) .....	18
2.9 PiTFT - 2.8" <i>Touchscreen Display</i> untuk Raspberry Pi .....	20
2.10 Ultrasonik HC-SR04 Sebagai Pengukur Jarak .....	22
2.11 Bekerja dengan Putty dan VNC .....	23
2.12 Memulai Image Processing dengan OpenCV .....	26
2.12.1 Dasar OpenCV dan bahasa C++ .....	26
2.12.2 Menampilkan gambar bentuk .....	28
2.12.3 Smoothing .....	31
2.12.4 Ambang Batas .....	31
2.12.5 Deteksi Tepi Canny .....	32

2.12.6	Lingkaran Hough.....	32
2.12.7	Kontur.....	33
BAB 3	PERANCANGAN SISTEM.....	37
2.13	Observasi dan Analisa Masalah.....	37
2.14	Studi Literatur.....	38
2.15	Perancangan Sistem.....	38
2.16	Pembuatan Alat.....	39
2.17	Perancangan Program dan Pengujian.....	40
2.18	Pengumpulan Data dan Evaluasi .....	43
2.19	Penyusunan Laporan.....	43
BAB 4	PENGUJIAN DAN ANALISIS SISTEM .....	45
4.1	Pengujian Fungsi <i>Hardware</i> .....	45
4.1.1	Menampilkan Raspbian .....	45
4.1.2	Konfigurasi Pi TFT.....	45
4.1.3	Tes Kamera Raspberry Pi ( <i>RaspiCam</i> ).....	46
4.1.4	Konfigurasi <i>OpenCV</i> dan CodeBlocks pada <i>Raspberry Pi</i> .....	46
4.2	Pengujian Algoritma.....	46
4.2.1	Pengamatan Objek Uji Coba .....	47
4.2.2	Kalibrasi Jarak Awal 30 cm.....	49
4.2.3	Pengujian Algoritma Hough Circle .....	49
4.2.4	Pengujian Deteksi <i>Contours</i> dan <i>Bounding Box</i> .....	50
4.2.5	Pengujian Sistem pada <i>Prototype</i> .....	50
4.2.6	Menemukan Nilai Diameter Terkecil .....	52
4.3	Pengujian Keandalan Alat Menggunakan Ultrasonik SR04 .....	53
4.3.1	Kalibrasi Sistem dengan Sensor Ultrasonik.....	53
4.3.2	Pengujian Dinamisasi Jarak dengan Ultrasonik.....	55
4.4	Pengujian Sistem pada Kayu Nyata Industri.....	57
4.4.1	Ketahanan Power Supply.....	60
4.4.2	Portabilitas Alat .....	60
4.4.3	Kemudahan Duplikasi Alat .....	60
4.4.4	Potensi Pengguna.....	60
4.4.5	Analisa Fungsi Sistem .....	61
BAB 5	PENUTUP .....	63
4.5	Kesimpulan.....	63
4.6	Saran .....	64
	DAFTAR PUSTAKA .....	65
	LAMPIRAN .....	67
	BIODATA PENULIS .....	91

## DAFTAR TABEL

Tabel 2.1 Pheriperal dasar raspberry pi.....	9
Tabel 2.2 Tabel perintah untuk RaspiCam .....	19
Tabel 2.3 Struktur untuk point, size, rect, dan scalar .....	27
Tabel 4.1 Rekapitulasi Pengujian Hardware Raspberry Pi 2 .....	47
Tabel 4.2 Hasil Pengujian kehandalan alat pada <i>prototype</i> .....	50
Tabel 4.3 Tabel hasil kalbrasi ultrasonik dan kamera pada objek grade 15 .....	54
Tabel 4.4 Hasil pengujian sistem dengan ultrasonik terhadap tiga buah kayu bundar grade 15. Tabel dengan warna dasar merah muda nilainya diabaikan karena nilai tersebut merupakan perpindahan posisi.....	55
Tabel 4.5 Hasil Pengujian pada Kayu Nyata Industri .....	58

*Halaman ini sengaja dikosongkan*

## DAFTAR GAMBAR

Gambar 2.1 Pengukuran diameter kayu konvensional menggunakan alat bantu meteran. Tampak gambar kiri pengukuran diameter ujung, dan tampak kanan pengukuran diameter pangkal .....	6
Gambar 2.2 Model Raspberry P, komputer kecil yang dapat digunakan sebagai inti sistem. ....	8
Gambar 2.3 Logo Raspbian.....	10
Gambar 2.4 Tampilan Win32 Disk Imager .....	11
Gambar 2.5 Raspberry Pi Software Configuration Tool (raspi-config) ..	11
Gambar 2.6 Tampilan Chos boot option .....	12
Gambar 2.7 Logo Software OpenCV .....	13
Gambar 2.8 Code::Blocks .....	16
Gambar 2.9 Tampilan konfigurasi compiler codeblocks .....	16
Gambar 2.10 Tampilan konfigurasi linker CodeBlocks.....	16
Gambar 2.11 Tampilan tambahan <i>linker option</i> .....	17
Gambar 2.12 Pemilihan console baru (kiri) dan pemilihan bahasa (kanan).....	18
Gambar 2.13 Kamera RaspiCam.....	18
Gambar 2.14 PiTFT - 2.8" Touchscreen Display untuk Raspberry Pi ..	20
Gambar 2.15 Tampilan awal terminal mengunduh data <i>repository adafruit</i> .....	21
Gambar 2.16 Ultrasonik HC-SR04 .....	22
Gambar 2.17 Rangkaian pembagi tegangan.....	23
Gambar 2.18 Tampilan Raspi-config untu meng- <i>enable</i> SSH server ...	24
Gambar 2.19 Konfigurasi awal VNC viewer .....	25
- Langkah 3: Sebelum dapat terhubung ke Pi menggunakan VNC, harus terlebih dahulu memulai VNC server. SSH menyediakan mekanisme ini. Untuk mulai dengan, kita mengambil Client SSH kami - Putty. Setelah Anda download, hanya memulai - itu adalah exe mandiri dan ditampilkan pada gambar 2.20 .....	25
Gambar 2.21 Konfigurasi Putty .....	26
Gambar 2.22 Tampilan halaman login Putty .....	26
Gambar 2.23 Struktur dasar OpenCV .....	27
Gambar 2.24 Meskipun OpenCV diimplementasikan dalam C, struktur yang digunakan dalam OpenCV berorientasi pada objek desain sehingga, IplImage berasal dari CvMat, yang berasal dari CvArr .....	28
Gambar 2.25 Hasil dari berbagai jenis ambang batas di cvThreshold (). Garis horizontal melalui setiap grafik merupakan ambang batas	



tertentu diterapkan pada grafik atas dan efeknya untuk masing-masing lima jenis operasi ambang bawah .....	31
Gambar 2.26 Hasil Canny edge detection untuk dua gambar ketika diberi pengaturan batas tinggi dan rendah.....	32
Gambar 2.27 Persegi sebagai <i>bounding rect</i> .....	35
Gambar 2.28 Kontur berupa sepuluh poin dengan melampirkan lingkaran minimal (a) dengan ketepatan ellipsoid (b); kotak (c) digunakan oleh OpenCV untuk mewakili ellipsoid .....	35
Gambar 3.1 Flowchart pelaksanaan kegiatan .....	37
Gambar 3.2 Industri pengolahan kayu PT. Indofurnitama Raya, Pasuruan .....	38
Gambar 3.3 Rancangan sistem utama .....	38
Gambar 3.4 Gambaran alat yang diterapkembangkan .....	39
Gambar 3.5 Keterangan gambaran alat yang diterapkembangkan.....	40
Gambar 3.6 Visualisasi uji coba devais saat melakukan pengukuran terhadap.....	40
Gambar 3.7 Flowchart alur bagian <i>hardware</i> sistem .....	41
Gambar 3.8 Flowchart algoritma <i>software</i> sistem .....	42
Gambar 4.1 Frame kayu yang akan diujicoba dengan berbagai ukuran, 19 cm (tampak kiri) dan 18 cm (tampak kanan).....	45
Gambar 4.2 Pengujian kamera raspberry pi (tampak kiri) dan hasil yang ditampilkan kamera (tampak kanan) .....	46
Gambar 4.3 Proses <i>build</i> dan <i>run</i> pada CodeBlocks berjalan dengan baik .....	46
Gambar 4.4 Gambar prototype A dengan grade 18 .....	47
Gambar 4.5 Gambar prototype B dengan grade 14 .....	47
Gambar 4.6 Gambar prototype C dengan grade 19 .....	48
Gambar 4.7 Gambar nyata kayu yang diuji dengan grade 19 .....	48
Gambar 4.8 Gambar nyata kayu B yang diuji dengan grade 18 .....	48
Gambar 4.9 Gambar nyata kayu B yang diuji dengan grade 17 .....	48
Gambar 4.10 Tampilan pegujian program menggunakan algoritma <i>hough circle</i> . Nilai yang dihasilkan memiliki error s.d. 11% (terlalu besar).....	49
Gambar 4.11 <i>Screenshot</i> pengujian terhadap prototype A (grade 18) ..	51
Gambar 4.12 <i>Screenshot</i> pengujian terhadap prototype B (grade 14) ..	51
Gambar 4.13 <i>Screenshot</i> pengujian terhadap prototype C (grade 19) ..	52
Gambar 4.14 Sistem menentukan diameter terkecil pada prototype (Status: Akurat) .....	52

Gambar 4.15 Sistem menentukan diameter terkecil pada prototype (Status: Akurat) .....	52
Gambar 4.16 Sistem menentukan diameter terkecil pada prototype (Status: Akurat) .....	53
Gambar 4.17 Objek Pengujian sistem pada kayu grade 15 (Gambar kanan) dan nilai sebenarnya melalui pengukuran konvensional. Diameter terkecil ditunjukkan oleh garis oranye dengan nilai 15,07 cm .....	53
Gambar 4.18 Kalibrasi ultrasonik SR04 terhadap kamera dengan jarak 40cm, 30 cm, dan 20 cm.....	54
Gambar 4.19 Objek kayu uji dengan grade 15 .....	55
Gambar 4.20 Grafik pengujian ultrasonik terukur tanpa nilai perpindahan. ....	57
Gambar 4.21 Screenshot pengujian terhadap kayu A (grade 17).....	58
Gambar 4.22 Screenshot pengujian terhadap kayu B (grade 18) .....	59
Gambar 4.23 Screenshot pengujian terhadap kayu C (grade 19) .....	59
Gambar 4.24 Outline metode .....	61
Gambar 0.1 Tampilan penampang sistem .....	67
Gambar 0.2 Tampilan penampang kayu .....	67

*Halaman ini sengaja dikosongkan*

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Hutan merupakan salah satu sumber daya alam yang penting di Indonesia dan memberikan manfaat langsung dan tidak langsung. Manfaat langsung antara lain berupa kayu yang dapat dipanen dan diolah. Salah satu hasil dari kayu yang dipanen adalah berupa kayu gelondong atau biasa disebut kayu bulat. Kayu tersebut merupakan bahan dasar dari produk jadi seperti lemari, kursi dan lain sebagainya. Akan tetapi terdapat permasalahan yang dihadapi oleh pelaku usaha kayu, yakni penentuan harga dari kayu gelondong yang dibeli. Pada umumnya penentuan harga kayu dilakukan setelah melakukan pengukuran pada diameter dan panjang kayu tersebut, sehingga dapat dihitung isi kayu dengan nilai kubisasinya.

Permasalahan utama yang dihadapi termasuk pada industri kayu, yaitu terdapat potensi kesalahan pada saat pengukuran dikarenakan diameter kayu yang tidak berbentuk lingkaran sempurna, sehingga kesalahan satu centimeter saja sangat berpengaruh terhadap harga kayu gelondong tersebut. Persepsi dalam penentuan nilai dari diameter terkecil kayu sebagai langkah awal dari perhitungan yang dilakukan satu orang dengan yang lain adalah berbeda. Sehingga kesalahan atau *error* pengukuran yang terjadi dapat menimbulkan kerugian dari pihak industri karena volume yang dibayar tidak sesuai dengan kondisi asli volume kayu tersebut.

Selama ini alat bantu yang digunakan hanya menggunakan meteran konvensional yang kurang presisi. Potensi kerugian pada mitra akibat salah memasukkan nilai salah satu *grade low* mencapai Rp. 40.000,00 per centimeter kayu bundar. Dari data yang diambil dari salah satu industri kayu di Pasuruan, Indo Furnitama Raya, Terdapat sekitar 12.000 kayu dalam sekali produksi sehingga didapatkan potensi nilai kerugian mencapai Rp 480.000.000,00. Padahal untuk *grade high* nilainya bisa lebih besar dari itu per centimeter kayu bundar tergantung jenis kayu yang digunakan.

Sehingga dibutuhkan suatu penyelesaian berupa inovasi alat. Pengolahan citra merupakan teknologi tepat guna untuk memberikan solusi terhadap permasalahan tersebut. Penelitian menggunakan algoritma pengolahan citra telah dilakukan [1], [2], [3], [6] maupun dengan

spesifikasi tentang permukaan kayu [10]. Teknologi yang dirancang memanfaatkan penginderaan mesin visual yang dilakukan oleh kamera untuk melakukan penghitungan otomatis terhadap nilai kayu.

## 1.2 Perumusan Masalah

Permasalahan yang coba dicari penyelesaiannya melalui tugas akhir ini adalah:

1. Bagaimana melakukan pengukuran kayu secara tepat dengan menggunakan penginderaan visual untuk mengatasi masalah pada industri kayu?
2. Bagaimana mendapatkan hasil pengukuran dengan tingkat presisi yang tinggi dengan kesalahan atau eror yang kecil melalui algoritma program pendukung?
3. Bagaimana merancang devais alat bantu pengukuran kayu menggunakan aplikasi *raspberry pi* sebagai antarmuka?

## 1.3 Tujuan Penelitian

Tujuan yang ingin dicapai dari program kami ini adalah:

1. Memperoleh hasil pengukuran yang presisi tanpa melakukan kontak fisik dengan kayu yang diukur melalui teknologi *machine vision* untuk mengatasi masalah kerugian pada industri kayu.
2. Menggunakan teknologi penginderaan visual pada industri kayu secara tepat guna.
3. Membuat devais alat bantu pengukuran kayu menggunakan *raspberry pi* sebagai antarmuka.

## 1.4 Batasan Masalah

Batasan masalah dari tugas akhir ini adalah sebagai berikut:

1. Menggunakan devais yang menunjang kinerja *pengolahan citra*.
2. Membuat alat pengukur yang portabel.
3. Mampu menampilkan nilai diameter terukur dalam suatu tampilan antarmuka.
4. Menampilkan nilai diameter terukur ditampilkan dalam satuan centimeter.

## 1.5 Metodologi Penelitian

Dalam tugas akhir ini, metodologi penelitian yang digunakan adalah sebagai berikut:

### 1. Observasi dan Analisa Masalah

Pada tahap ini dilakukan pengamatan terhadap permasalahan, situasi dan kondisi yang terjadi pada proses pengukuran di industri kayu, permasalahan selanjutnya akan dianalisa dan nantinya akan dipadukan dengan teknologi pada alat.

### 2. Studi Literatur

Studi literatur berisi serangkaian pengkajian sumber-sumber yang relevan dan terpercaya dalam pengumpulan materi serta menjadi acuan dalam penulisan. Digunakan banyak literatur berbeda agar dapat menghasilkan informasi yang lengkap.

### 3. Perancangan Sistem

Perancangan alat ini bertujuan untuk mendapatkan desain dan mekanisme yang optimal dengan memperhatikan data yang telah didapat dari studi literature dan observasi. Adapun rancangan kayu meter sebagaimana ditunjukkan pada lampiran gambaran teknologi yang akan diterapkembangkan, dengan desain pada.

### 4. Pembuatan Alat

Pembuatan alat dilakukan dengan dasar dari tahap perancangan alat yang sudah matang. Pembuatan alat diawali dengan pemrograman visual dengan menggunakan *software OpenCV* dan *CodeBlocks*, lalu setelah itu pengintegrasian antara program dengan hardware yakni dengan Raspberry pi, dengan mengandalkan kinerja *machine vision* maka hasil pengukuran yang presisi akan didapat dan akan ditampilkan pada layar lcd (*screen*).

### 5. Perancangan Program dan Pengujian

Pengujian dilakukan dengan mengukur objek kayu bundar berupa prototype kayu atau kayu bundar yang ntara dan akan dievaluasi hasil pengukuran apakah sudah sesuai dengan hasil sesungguhnya.

### 6. Pengumpulan data dan evaluasi

Data yang diperoleh dari hasil Pengukuran yang dilakukan secara berulang ulang akan segera di kumpulkan datanya dan akan dievaluasi mengenai kepresisian data yang dihasilkan.

### 7. Penyusunan Laporan

Tahap akhir dari rangkaian ini adalah penyusunan dan pembuatan sebagai laporan.

Selain itu penyusunan laporan dapat digunakan sebagai bahan referensi dan dokumentasi penyelesaian sistem.

## 1.6 Sistematika Penulisan

Laporan tugas akhir ini terdiri dari Lima Bab dengan sistematika penulisan sebagai berikut:

### **Bab 1 Pendahuluan**

Bab ini meliputi latar belakang, perumusan masalah, tujuan penelitian, batasan masalah, metodologi, sistematika penulisan, dan relevansi.

### **Bab 2 Dasar Teori**

Bab ini menjelaskan tentang dasar-dasar teori yang dibutuhkan dalam pengerjaan tugas akhir ini, yang meliputi dasar pengaturan hardware dan software raspberry pi, dasar sensor ultrasonik, dasar interaksi manusia mesin, dan algorithm pengolahan citra yaitu lingkaran Hough, deteksi tepi Canny, kontur, dan *bounding box*.

### **Bab 3 Perancangan Sistem**

Bab ini menjelaskan tentang perancangan desain, rangkaian elektronik, serta perangkat lunak. Bab ini juga berisi menjelaskan tentang prosedur pengujian yang dilakukan dalam penelitian.

### **Bab 4 Pengujian dan Pembahasan Sistem**

Bab ini menjelaskan tentang hasil yang didapat dari pengujian kehandalan sistem sebagai alat pengukur kayu.

### **Bab 5 Penutup**

Bab ini menjelaskan tentang kesimpulan meliputi kekurangan-kekurangan pada kerja alat dari hasil analisa serta saran untuk pengembangan ke depan.

## 1.7 Relevansi

Hasil dari tugas akhir ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Menjadi alat standarisasi pengukuran diameter kayu di Indonesia.
2. Mendapatkan nilai terbaik dari diameter kayu untuk mengurangi rugi akibat error pada proses pengukuran kayu.
3. Pengembangan dari ilmu pengolahan citra digital.

## **BAB 2**

### **TINJAUAN PUSTAKA**

#### **2.1 Diameter Terkecil Kayu**

Perhitungan pada sistem dilakukan terhadap diameter terkecil. Diameter terkecil secara visual dapat diketahui dengan melihat penampang permukaan kayu bundar, yaitu bagian lingkaran tahun terluar dari kayu. Terbentuknya lingkaran tahun [14] pada sebuah batang pohon dimulai pada saat pohon pertama kali tumbuh hingga ditebang atau mati. Lingkaran tahun ini terbentuk hanya satu kali untuk setiap tahunnya dan jumlahnya akan terus bertambah pada tahun-tahun berikutnya selama pohon itu tumbuh. Semakin tua usia sebuah pohon maka semakin banyak jumlah lingkaran tahunnya. Oleh karena itu umur sebuah pohon dapat diketahui melalui jumlah lingkaran tahun yang ada pada batangnya. Masing- masing jenis kayu memiliki karakteristik pertambahan ukuran tersendiri untuk setiap tahun.

Deteksi yang dilakukan adalah sistem harus mampu mengenali kontur yang dihasilkan antara lingkaran tahun terluar yang berbatasan dengan kulit terluar kayu dan ditandai oleh kayu gubal. Kayu gubal adalah bagian pinggir atau tepi pada sebuah kayu pada semua jenis kayu baik itu kayu jati, mahoni, sonokeling, kayu eboni, mindi dll. Umumnya, kayu gubal berwarna terang atau putih, lebih terang dari pada kayu teras. Kayu teras adalah bagian tengah atau inti dari sebuah kayu. Pada jenis kayu sengon, kayu teras dan kayu gubal hampir tidak memiliki perbedaan warna sehingga tidak mengganggu proses deteksi kontur. Jika pengukuran dilakukan terhadap kayu yang memiliki perbedaan kontur dan warna yang kuat, maka pengaturan ambang batas perlu dilakukan.

#### **2.2 Perhitungan Diameter Kayu Secara Konvensional**

Berdasarkan PP N0. 28 Tahun 1985, tentang Perlindungan Hutan Pasal 13, dinyatakan bahwa untuk melindungi hak-hak negara yang berkenaan dengan hasil hutan, maka terhadap semua hasil hutan harus diadakan pengukuran dan pengujian [12]. Hal tersebut dijadikan sebagai dasar. perhitungan penetapan besarnya pungutan negara. Adapun kegunaan lainnya dari hasil pengukuran tersebut adalah sebagai dasar perhitungan/harga jual/penjualan, laba rugi perusahaan, upah buruh, penyusunan statistik dan lain sebagainya. Karena pentingnya data hasil



pengukuran tersebut, maka diperlukan pengetahuan tentang bagaimana cara melaksanakan pengukuran dan pengujian hasil hutan.

Bagi pekerja perkayuan di Indonesia, cara yang biasa dilakukan adalah dengan mengukur secara manual menggunakan alat bantu meteran untuk didapatkan data berupa diameter dan juga panjang. Secara matematis, perhitungan isi kayu bundar menggunakan rumus Brereton, yaitu dengan rumus:

$$Isi = \frac{0,7954 \times (dp + du) \times p}{10000} \quad (2.1)$$

Keterangan:

Isi	= Isi kayu bundar dalam satuan m <sup>3</sup>
0,7854	= Konstanta Brereton (nilai kubisasi)
dp	= Diameter pangkal
du	= Diameter ujung
p	= panjang

Perhitungan mengenai isi kayu bundar (log) terlebih dahulu dicari diameter rata-rata antara diameter pangkal dengan diameter ujung. Padahal lingkaran luar kayu memiliki bentuk yang berbeda-beda dan tidak berbentuk bulat sempurna, sehingga perhitungan tersebut memiliki potensi *error* yang sangat besar.



*Gambar 2.1 Pengukuran diameter kayu konvensional menggunakan alat bantu meteran. Tampak gambar kiri pengukuran diameter ujung, dan tampak kanan pengukuran diameter pangkal*

## 2.3 Penginderaan Visual

*Machine Vision* [3] adalah transformasi data dari gambar atau video ke baik berupa logika keputusan atau menampilkan representasi baru. Secara umum bidang ini dapat mengolah informasi dari keadaan real (visualisasi) berupa pengolahan citra agar didapatkan model dari proyeksi-proyeksi benda yang mana memerlukan banyak inversi untuk mendapatkan suatu pemetaan yang mendekati bentuk asli suatu benda, yang kemudian diidentifikasi sesuai kebutuhan.

Dalam pengolahan citra, proses segmentasi adalah hal yang sangat penting. Dalam hal ini dilakukan proses pemetaan informasi citra yang diperoleh dalam beberapa kategori. Dalam hal ini pemrosesan tersebut dilakukan dengan memanfaatkan beberapa hal diantaranya pixel, dan warna. Setiap warna memiliki nilai dan menjadikan satu warna dan warna lain bernilai berbeda. Demikian juga dengan penampakan warna hitam- putih (*grayscale*). Warna diantara hitam dan putih adalah warna abu-abu dengan derajat nilai yang berbeda-beda dari 0-255. Media yang digunakan untuk dapat melakukan proses tersebut adalah kamera.

## 2.4 Interaksi Manusia- Mesin

Untuk dapat menggunakan perangkat, diperlukan suatu *interface* (antarmuka) agar pengguna dapat berinteraksi dengan komputer/ mesin. Dalam *interface* ini dilakukan perancangan, evaluasi, dan implementasi antarmuka pengguna dengan suatu media misalnya komputer agar mudah digunakan oleh manusia. Diungkapkan oleh Jun Su Ha (2013) [2] bahwa interaksi manusia dan komputer sendiri adalah serangkaian proses, dialog dan kegiatan yang dilakukan oleh manusia untuk berinteraksi dengan komputer yang keduanya saling memberikan masukan dan umpan balik melalui sebuah antarmuka untuk memperoleh hasil akhir yang diharapkan. Untuk mendapatkan data pengukuran akan diintegrasikan sebuah layar monitor dengan ukuran yang disesuaikan dengan papan perintah yang digunakan dan membaca status pengukuran yang terhubung ke kontrol pusat pada *raspberry pi 2*.

## 2.5 Perangkat Raspberry Pi sebagai Single Board Circuit

**Raspberry Pi**, sering juga disingkat dengan nama *Raspi* atau **RPI**, adalah komputer mini (*Single Board Circuit* /SBC) yang memiliki

ukuran sebesar kartu kredit [13]. Raspberry Pi bisa digunakan untuk berbagai keperluan, seperti *spreadsheet*, *game*, bahkan bisa digunakan sebagai *media player* karena kemampuannya dalam memutar *video high definition*. Raspberry Pi 2 yang digunakan merupakan generasi terbaru (medio Maret 2015) setelah A, A+, B, dan B+.

Raspberry Pi 2 bersifat *open source* (berbasis Linux), Raspberry Pi 2 bisa dimodifikasi sesuai kebutuhan penggunaannya. Sistem operasi utama Raspberry Pi menggunakan *Debian GNU/Linux* dan bahasa pemrograman *Python*. Sistem operasi tersebut dibuat berbasis Debian yang merupakan salah satu distribusi *Linux OS*.



Gambar 2.2 Model Raspberry P, komputer kecil yang dapat digunakan sebagai inti sistem.

Spesifikasi Raspberry pi 2:

- Catu daya : 5 VDC, 700 mA (via micro USB)
- Berbasis mikrokontroler
  - /mikroprosesor : ARM1176JZF-S core, 1 GHz.
- Jumlah port I/O : 8 pin GPIO
- Port antarmuka : UART TTL, SPI, I2C, USB, Composite RCA, 3.5 mm jack, 10/100q- Ethernet (RJ45), LCD Panels via DSI, CSI(Camera Serial Interface), HDMI.
- Bootloader : melalui OS berbasis LINUX.
- Port pemrograman : -
- Fitur : Memory 512 MB, 2 USB PORT, Graphics Broadcom VideoCore IV , SD Card Slot.
- Dimensi : 85.60mm(L) x 56mm(W) x 21mm(H)

### 2.5.1 Memulai Perangkat Raspberry Pi 2

Raspberry pi 2 atau RPi 2 dapat dikonfigurasi seperti konvensional komputer pada umumnya, yaitu dengan keyboard, mouse dan juga display. Untuk devais USB yang mengkonsumsi lebih dari 100mA, sangat dianjurkan untuk dihubungkan dengan catu daya tambahan.

*Tabel 2.1 Pheriperal dasar raspberry pi*

Item	Komputer Umum	Start Up Raspberry
Raspberry Pi board	Yes	Yes
Power Supply	Yes	Yes
SD card +OS	Yes	Yes
USB keyboard	Yes	No
USB mouse	Jika menggunakan GUI desktop	No
Display	Yes	No
Display cable	Yes	No
Kabel network	Jika terhubung koneksi	Kecuali model A, Yes
Powered USB hub	Yes	No

#### 2.5.1.1 Menyiapkan OS (Operating System) dalam SD Card

Pada dasarnya RPi tidak memiliki *storage* internal *built in* sehingga dibutuhkan SD card pada seri sebelum B+ dan menggunakan *micro SD* untuk seri terbaru RPi. Setelah dilakukan *install* terjadi initial *bootup* sebagai tahap awal memulai RPi.

#### 2.5.1.2 Keyboard dan Mouse

Hampir semua jenis *USB keyboard* dan *mice* beroperasi dengan baik pada RPi. Untuk jenis *wireless* juga berfungsi dengan baik dan hanya membutuhkan satu USB port. Sedangkan untuk *bluetooth keyboard* atau *mouse*, dibutuhkan bluetooth donle yang juga mebutuhkan satu port.

#### 2.5.1.3 Display

Ada dua jenis display raspberry yang dapat digunakan untuk mengaktifkan RPi sebagai primary monitor. Yaitu dengan HDMI (*High*

*definition*) dan *composite (low definiton)*. TV HDMI dan berbagai jenis monitor LCD dapat dihubungkan dengan RPi melalui kabel standar HDMI versi 1.3 dan 1.4. Sedangkan untuk *composite (Kabel kuning ke kuning)*, *PAL* dan *NTSC* juga mendukung RPi. Konverter dari HDMI ke VGA dapat digunakan untuk melengkapi penggunaan display untuk RPi.

#### 2.5.1.4 Kabel

Beberapa jenis kabel yang dapat digunakan untuk koneksi RPi antara lain: *Micro-B USB* sebagai kabel power, HDMI, Audio/video cable, Ethernet/LAN cable

## 2.6 Raspbian

Raspbian adalah suatu *free-operating system (OS)* yang berbasis Linux dan teroptimasi khusus untuk RPi [13]. OS ini digunakan untuk program basis dan *utility* menjalankan Raspberry Pi atau RPi.



Gambar 2.3 Logo Raspbian

Ketika RPi terinisialisasi, pengguna harus *log in* terlebih dahulu dimana sistem *default* dari raspbian yaitu **username: pi**, dan **password: raspberry**. Pengaturan awal setelah *boot up* adalah dengan perintah: **raspi-config**. Dalam perintah **raspi-config** dapat diatur tampilan awal menggunakan **GUI** (*Graphical User Interface*) atau *Command Line Interface*. Berikut akan dijelaskan langkah- langkah *start up* RPi dengan raspbian.

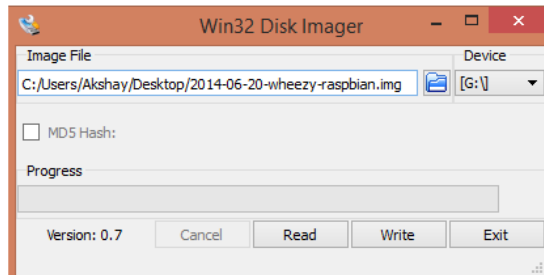
### 2.6.1 Mengunduh Raspbian dan Image Writer

Melalui komputer atau laptop pengguna, unduh versi raspbian terbaru melalui [www.raspberrypi.org/downloads/](http://www.raspberrypi.org/downloads/). Dibutuhkan *image writer* untuk menuliskan (*write*) OS yang telah terdownload ke dalam SD card/micro SD yang akan digunakan menggunakan salah satu aplikasi *open-source* yaitu **win32 disk imager**.

### 2.6.2 Menuliskan Image (Write Process)

Masukkan micro SD pada laptop atau PC dan saat terbuka lakukan pencarian pada file *raspbian image (\*.img)* kemudian pilih

devais *drive/local disk micro* SD dengan benar. Jika terjadi kesalahan dalam pemilihan akan menyebabkan kerusakan pada *local disk* pengguna sehingga harus berhati hati. Kemudian pilih tombol “write” pada bagian bawah tampilan *console*. Seperti pada gambar 2.4 ditunjukkan pemilihan image, dimana letak *drive* direpresentasikan pada letter “G:\”/



Gambar 2.4 Tampilan Win32 Disk Imager

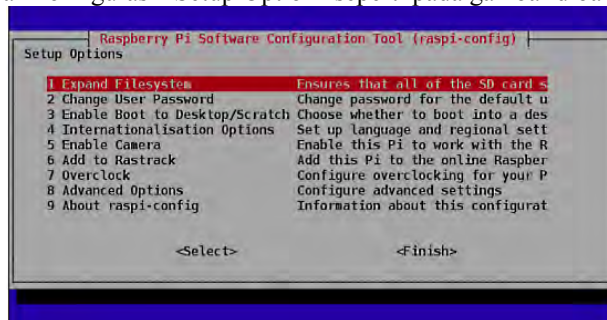
### 2.6.3 Penyetelan Pengaturan Awal Raspberry Pi

Setelah booting RPi akan ada situasi *user credentials* untuk memasukkan *username* dan *password* dengan mode *default*.

login: pi

password: raspberry

Ketika RPi sukses melakukan *booting* pertama kali akan dilakukan konfigurasi “Setup Option” seperti pada gambar dibawah.

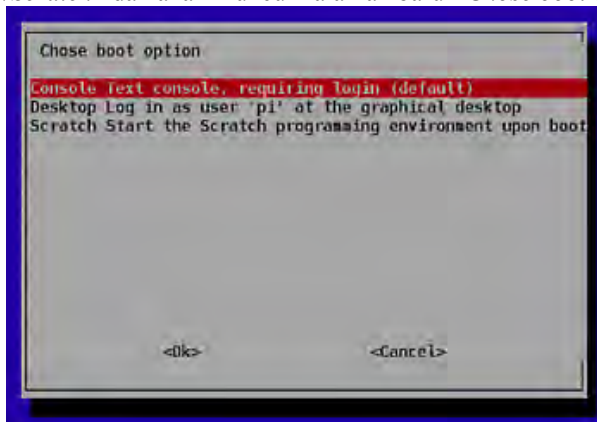


Gambar 2.5 Raspberry Pi Software Configuration Tool (raspi-config)

Halaman pengaturan dapat diakses melalui perintah:

Sudo raspi-config

Pada halaman pengaturan, yang pertama kali dilakukan adalah perintah **“Expand Filesystem”**, sehingga tampilan nilai disk yang kosong dapat ditampilkan oleh micro SD sebagai partisi penuh dari RPi. Langkah kedua adalah dengan memilih **“Enable Boot to Desktop/Scratch”** dan akan muncul halaman baru **“Chose boot option”**.



Gambar 2.6 Tampilan Chos boot option

Pada halaman tersebut digunakan pilihan kedua yaitu **“Desktop Login as user ‘pi’ at the graphical desktop”** dan tekan enter. Kemudian proses akan menampilkan kembali halaman **“Setup Option”**. Kemudian setelah menekan **“finish”**, sistem akan melakukan *reboot* secara otomatis.

## 2.7 Install OpenCV pada Raspberry Pi 2

OpenCV (*Open Computer Vision*) adalah sebuah API (*Application Programming Interface*), *Library* yang sudah sangat familiar pada pengolahan citra atau *machine vision* [4]. Dengan vision tersebut komputer dapat mengambil keputusan, melakukan aksi, dan mengenali terhadap suatu objek.

OpenCV merupakan open source (lihat <http://opensource.org>) yang dilengkapi library yang ditulis dalam C dan C ++ dan berjalan di Linux, Windows dan Mac OS X. Pengembangan aktif pada antarmuka untuk Python, Ruby, Matlab, dan bahasa lainnya.



Gambar 2.7 Logo Software OpenCV

OpenCV dirancang untuk melakukan efisiensi dalam komputasi, dan dengan fokus pada tampilan secara *realtime*. Rutinitas dioptimalkan dalam algoritmik. OpenCV secara otomatis menggunakan perpustakaan IPP sesuai pada perpustakaan yang telah diinstal. Salah satu tujuan OpenCV adalah untuk menyediakan infrastruktur visual komputer sederhana yang digunakan untuk membantu orang membangun aplikasi penginderaan visual.

Perpustakaan OpenCV berisi lebih dari 500 fungsi, termasuk pabrik inspeksi produk, pencitraan medis, keamanan, user interface, kalibrasi kamera, penglihatan stereo, dan robotika.

OpenCV adalah *library Open Source* untuk interaksi manusia mesin untuk C/C++. OpenCV didesain untuk aplikasi *realtime*, dan memiliki fungsi-fungsi akuisisi yang baik untuk gambar/video. Berikut langkah-langkah yang dilakukan untuk menginstal OpenCV pada Raspberry Pi 2 atau RPi2.

Langkah 0: Pada terminal, lakukan perintah update dan upgrade firmware RPi.

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo rpi-update
```

Langkah 1: Install developer tools dan packages yang diperlukan.  
(estimasi waktu instal RPi 2: < 40 detik)

```
$ sudo apt-get install build-essential cmake pkg-config
```

Langkah 2: menginstal *image I/O packages* agar dapat menampilkan berbagai macam file berformat JPEG, PNG, TIFF, dll. (estimasi waktu instal RPi 2: < 30 detik)

```
$ sudo apt-get install libjpeg8-dev libtiff4-dev libjasper-dev libpng12-dev
```



Langkah 3: Menginstal *GTK development library* yang digunakan untuk membuat GUI dan dibutuhkan untuk proses “highgui”. (estimasi waktu instal RPi 2: < 3 menit)

```
$ sudo apt-get install libgtk2.0-dev
```

Langkah 4: Menginstal *video I/O packages* yang digunakan untuk menampilkan video. (estimasi waktu instal RPi 2: < 30 detik)

```
$ sudo apt-get install libavcodec-dev libavformat-dev  
libswscale-dev libv4l-dev
```

Langkah 5: Install *libraries* yang digunakan untuk optimasi operasi dengan OpenCV. (estimasi waktu instal RPi 2: < 30 detik)

```
$ sudo apt-get install libatlas-base-dev gfortran
```

Langkah 6: Install *pip*. (estimasi waktu instal RPi 2: < 30 detik)

```
$ wget https://bootstrap.pypa.io/get-pip.py  
$ sudo python get-pip.py
```

Langkah 7: Install *virtualenv* dan *virtualenvwrapper*. Kemudian update “~/.profile”. (estimasi waktu instal RPi 2: < 2 menit)

```
$ sudo pip install virtualenv virtualenvwrapper  
# virtualenv and virtualenvwrapper  
export WORKON_HOME=$HOME/.virtualenvs  
source /usr/local/bin/virtualenvwrapper.sh  
$ source ~/.profile  
$ mkvirtualenv cv
```

Langkah 8: Install Python 2.7 development tools

```
$ sudo apt-get install python2.7-dev
```

Langkah 9.1: Download OpenCV dan *unpack*

```
$ wget -O opencv-2.4.10.zip  
http://sourceforge.net/projects/opencvlibrary/files  
/opencv-unix/2.4.10/opencv-2.4.10.zip/download  
$ unzip opencv-2.4.10.zip  
$ cd opencv-2.4.10
```

Langkah 9.2: Aktifkan *Build* (estimasi waktu instal RPi 2: < 1.5 menit). Kemudian compile OpenCV. Pastikan setelah compile OpenCV, pengguna berada pada “*cv*” *virtual environment*. (estimasi waktu instal RPi 2: < 2.8 jam).

```
$ mkdir build
$ cd build
$ cmake -D CMAKE_BUILD_TYPE=RELEASE -D
  CMAKE_INSTALL_PREFIX=/usr/local -D
  BUILD_NEW_PYTHON_SUPPORT=ON -D
  INSTALL_C_EXAMPLES=ON -D
  INSTALL_PYTHON_EXAMPLES=ON -D BUILD_EXAMPLES=ON
..
```

Langkah 10: Akhirnya, Install OpenCV. OpenCV seharusnya terinstall oada `/usr/local/lib/python2.7/site-packages` untuk memaksimalkan OpenCV dengan “cv” virtual environment jika menggunakan python, dilakukan integrasi tambahan. Namun jika menggunakan codeblocks, akan dilakukan pengaturan pada bagian selanjutnya. (estimasi waktu instal RPi 2: < 1 menit).

```
$ sudo make install
$ sudo ldconfig

$ cd ~/.virtualenvs/cv/lib/python2.7/site-packages/
$ ln -s /usr/local/lib/python2.7/site-packages/cv2.so
  cv2.so

$ ln -s /usr/local/lib/python2.7/site-packages/cv.py
  cv.py
```

Langkah 11: Pengujian OpenCV (test drive). Jika telah berfungsi dapat dipastikan OpenCV telah siap digunakan.

```
$ workon cv
$ python
>>> import cv2
>>> cv2.__version__
'2.4.10'
```

## 2.8 Menggunakan Code::Blocks sebagai Compiler

### 2.8.1 Install dan memulai CodeBlocks

Untuk melakukan compile, digunakan software open-source Code::Blocks yang dapat digunakan dengan *environment OpenCV* dan dapat beroperasi dengan baik pada Linux.



Gambar 2.8 Code::Blocks

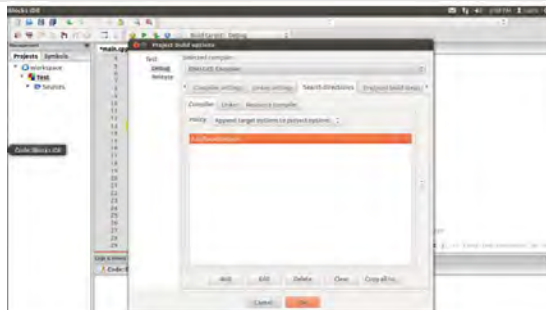
Beberapa konfigurasi dilakukan untuk memaksimalkan penggunaan dari *library OpenCV* yang telah terinstall.

Langkah 1: Buka CodeBlocks dan buat C++ project.

Langkah 2: Klik kanan pada project dan pilih “Build Options”

Langkah 3: pada halaman yang muncul, pilih “Search directories tab”, pilih pada “Compiler tab” dan add OpenCV “Include directory”. Tambah berupa:

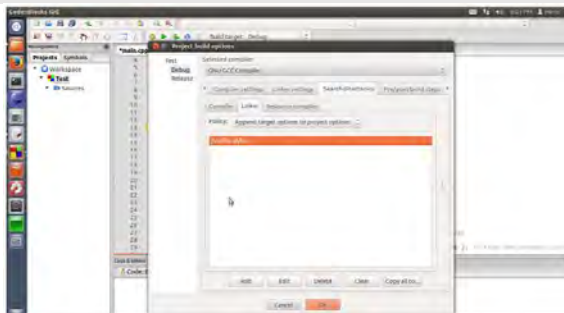
`/usr/local/include`



Gambar 2.9 Tampilan konfigurasi compiler codeblocks

Langkah 4: Buka “Linker tab” dan atur “OpenCV bin directory”

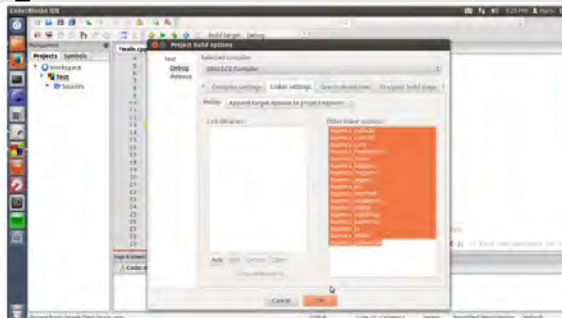
`/usr/local/bin`



Gambar 2.10 Tampilan konfigurasi linker CodeBlocks

Langkah 5: Buka “Linker setting” dan pada “Other linker options” tambahkan list libraries:

```
-lopencv_calib3d  
-lopencv_contrib  
-lopencv_core  
-lopencv_features2d  
-lopencv_flann  
-lopencv_highgui  
-lopencv_imgproc  
-lopencv_legacy  
-lopencv_ml  
-lopencv_nonfree  
-lopencv_objdetect  
-lopencv_photo  
-lopencv_stitching  
-lopencv_superres  
-lopencv_ts  
-lopencv_video  
-lopencv_videostab
```

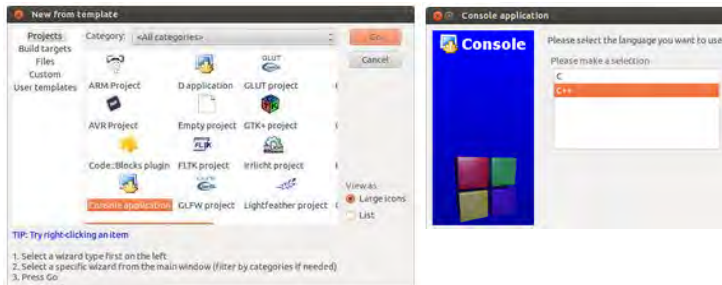


*Gambar 2.11 Tampilan tambahan linker option*

## 2.8.2 Memulai project baru CodeBlocks

Untuk memulai project baru pada CodeBlocks, setelah pengaturan pengaturan build option, dilakukan dengan beberapa langkah antara lain:

- Langkah 1: Pada bagian “create a new project” pilih Console application kemudian pilih bahasa yang digunakan C atau C++



Gambar 2.12 Pemilihan console baru (kiri) dan pemilihan bahasa (kanan)

- Langkah 2: Pilih compiler yang digunakan dan akhirnya CodeBlocks siap digunakan.

## 2.9 Raspberry Pi Camera (RaspiCam)

Untuk melakukan *image processing*, dibutuhkan kamera dengan resolusi yang baik karena pemanfaatan alat pengukur pada lapangan dilakukan baik didalam maupun luar ruangan. Penggunaan kamera difungsikan sebagai input untuk mendapatkan gambar dari kayu. Kamera yang digunakan adalah *Raspberry Pi Camera (RaspiCam)* yang dapat digunakan untuk mengambil gambar *High Definition (HD)* dengan fokus lima megapixel yang *support mode video 1080p30, 72p60* dan *VGAA90*.

RaspiCam bekerja dengan semua model Raspberry atau RPi yang dapat diakses melalui MMAL atau V4L APIs, dan berbagai *libraries* yang dibuat untuk RaspiCam [13].



Gambar 2.13 Kamera RaspiCam

Dihubungkan dengan 15 lajur *ribbon cable*, RaspiCam dapat langsung dikoneksikan dengan RPi. Pengaturan RaspiCam dengan RPi dapat dilakukan dengan beberapa langkah yaitu:

Langkah 1: Perbarui *kernel* terbaru, firmware, dan aplikasi.  
Dibutuhkan koneksi internet untuk memperbarui.

```
$ sudo apt-get update  
$ sudo apt-get upgrade
```

Langkah 2: Pilih “enable camera support” dengan mengetikkan *raspi-config* pada terminal. Kemudian reboot sistem.

```
$ sudo raspi-config
```

Langkah 3: Lakukan pengecekan (*test drive*) dengan perintah:

```
$ raspistill -v -o test.jpg
```

Langkah 4: Instal UV4L sebagai “driver” yang mempermudah penggunaan camera.

```
$ curl http://www.linux  
projects.org/listing/uv4l_repo/lrkey.asc | sudo  
apt-key add -  
$ sudo apt-get install uv4l-raspicam-extras  
$ sudo service uv4l_raspicam restart  
$ sudo rpi-update
```

Langkah 5: driver pilihan tambahan yang bisa diinstal antara lain:

```
$ sudo apt-get install uv4l-server  
$ sudo apt-get install uv4l-uvic  
$ sudo apt-get install uv4l-xscreen  
$ sudo apt-get install uv4l-mjpegstream  
$ sudo apt-get install uv4l-webrtc
```

Beberapa perintah yang umum digunakan untuk RaspiCam:

*Tabel 2.2 Tabel perintah untuk RaspiCam*

Perintah	Fungsi
<i>Raspistill</i>	Mengambil gambar
<i>Raspivid</i>	Mengambil video
<i>Time-lapse</i>	Mengambil gambar dengan interval dan menyatukan menjadi satu video
<i>raspiyuv</i>	Mengambil gambar dan menghasilkan <i>raw</i> file gambar yang belum terproses

## 2.10 PiTFT - 2.8" Touchscreen Display untuk Raspberry Pi

Monitor menjadi bagian penting untuk menampilkan hasil berupa pembacaan yang dibutuhkan. Melalui monitor pula *raspberry pi* atau RPi dapat menampilkan *image processing*. Fitur layar 2,8 " dengan 320x240 16 - bit pixel warna dan *overlay* sentuh resistif . Koneksi menggunakan antarmuka SPI kecepatan tinggi di RPi dan dapat menggunakan layar mini sebagai konsol, X pelabuhan jendela, ataupun langsung menampilkan gambar atau video. Selain itu dapat langsung dihubungkan pada tepat pada bagian atas RPi. PiTFT - 2.8" Touchscreen Display untuk Raspberry Pi dirancang untuk cocok dengan baik ke Pi Model A atau B tetapi juga bekerja baik-baik saja dengan Pi 2 atau Model B + selama pengguna tidak keberatan overhang PCB port USB dengan 5mm.



Gambar 2.14 PiTFT - 2.8" Touchscreen Display untuk Raspberry Pi

PiTFT memerlukan dukungan *kernel* [13] dan beberapa hal lain untuk membuat tampilan dapat berjalan dengan baik.

Langkah 1: Download kernel dari adafruit (adafruit.com). Ini merupakan perintah pertama untuk memasukkan apt.adafruit.com ke repository RPi sehingga aplikasi yang diinginkan dapat diambil langsung dari server. Perintah kemudian dilanjutkan untuk mengunduh dan menginstall yang PiTFT dibutuhkan. (Estimasi proses pada RPi 2: 20 menit)

```
$ curl -SLs https://apt.adafruit.com/add | sudo bash
$ sudo apt-get install -y adafruit-pitft-helper
```

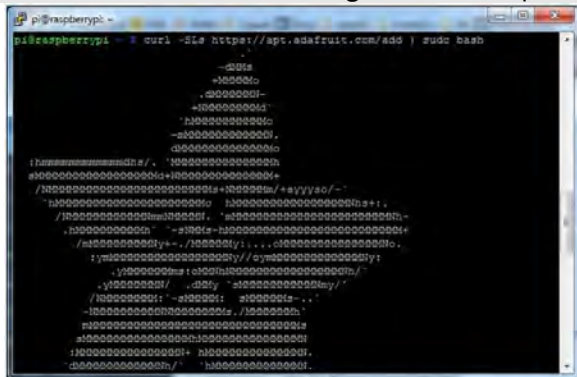
Langkah 2: Setelah kernel diinstal jalankan konfigurasi kernel. Menggunakan PiTFT helper akan memudahkan konfigurasi dengan mode *default*. Pada bagian akhir pengguna akan diminta pada apakah konsol teks muncul di PiTFT. Pilih Y atau N.

```
$ sudo adafruit-pitft-helper -t 28r
```

Langkah 3: PiTFT dapat digunakan sebagai *primary display* dan pada umumnya monitor yang terhubung dengan kabel HDMI menjadi monitor utama yang digunakan. Raspberry pi tidak dapat ditampilkan pada dua monitor utama dan harus memilih satu, sehingga harus dikonfigurasi agar PiTFT dijadikan layar utama.

Edit konfigurasi file dengan:

```
$ sudo mv /usr/share/X11/xorg.conf.d/99-fbturbo.conf
#atau
$ sudo nano /usr/share/X11/xorg.conf.d/99-pitft.conf
```



Gambar 2.15 Tampilan awal terminal mengunduh data repository adafruit

Langkah 4: Edit file seperti perintah dibawah. Setelah selesai tekan padakeyboard Ctrl+O dan enter.

Section "Device"

Identifier "Adafruit PiTFT"

Driver "fbdev"

Option "fbdev" "/dev/fb1"

EndSection

Ketikkan untuk memilih PiTFT sebagai layar utama:

Option "fbdev" "/dev/fb1"

Ketikkan untuk memilih monitor HDMI sebagai layar utama:

Option "fbdev" "/dev/fb0"

Langkah di atas akan membuat file konfigurasi yang memberi perintah jendela X yang harus menggunakan PiTFT framebuffer (terletak di / dev / FB1 ) secara default ketika berjalan.



Pada titik ini dapat digunakan alat “Raspi-config” untuk memungkinkan booting ke desktop seperti biasa pada Pi .

Jalankan perintah berikut :

```
sudo raspi-config
```

Kemudian memilih Enable Boot ke Desktop opsi / Scratch dan memilih jika Anda ingin boot ke konsol , desktop, atau lingkungan awal. Setelah keluar alat dan *reboot* akan ditampilkan Pi X jendela pada PiTFT (setelah estimasi sekitar 30 detik).

## 2.11 Ultrasonik HC-SR04 Sebagai Pengukur Jarak

HC-SR04 Ultrasonic sensor [13] untuk pengujian kehandalan dengan Raspberry Pi memiliki empat pin: *ground* (GND), *Echo Pulse Output* (ECHO), Pemicu Pulse Input (TRIG), dan catu daya 5V (Vcc). menggunakan Raspberry Pi untuk mengirim sinyal input untuk TRIG, yang memicu sensor untuk mengirim pulsa ultrasonik. Gelombang pulsa memantul dari benda di depannya dan kembali ke sensor. Sensor mendeteksi gelombang kembali ini dan mengukur waktu antara pemicu dan kembalinya sinyal tersebut, sehingga kemudian mengirim sinyal 5V pada pin ECHO.

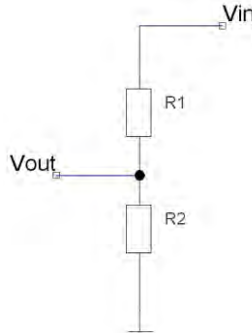


Gambar 2.16 Ultrasonik HC-SR04

ECHO akan bernilai "rendah" (0V) hingga sensor dipicu ketika menerima pulsa gema. Setelah pulsa yang dikirim oleh pemicu kembali, ECHO diatur bernilai "tinggi" (5V) dalam suatu durasi pulsa. Durasi pulsa adalah waktu antara keluaran dari bagian pemicu pulsa ultrasonik, dan terdeteksinya kembali oleh penerima sensor. Sinyal output sensor (ECHO) pada HC-SR04 dinilai pada 5V. Namun, pin input pada Raspberry Pi GPIO bernilai 3.3V. Mengirimkan sinyal 5V ke dalam tanpa pelindung ke 3.3V bisa merusak pin GPIO, oleh karena itu dibuat rangkaian pembagi tegangan yang terdiri dari dua resistor, untuk menurunkan tegangan output sensor.

### 2.11.1 Perhitungan Matematis

Rangkaian pembagi tegangan terdiri dari dua resistor (R1 dan R2) dalam seri terhubung dengan tegangan input (Vin), yang perlu dikurangi untuk tegangan output (Vout) . Dalam rangkaian, tegangan input ke raspberry dari ECHO, dikurangi dari 5V ke 3.3V.



Gambar 2.17 Rangkaian pembagi tegangan

Rumus rangkaian pembagi tegangan:

$$V_{out} = V_{in} \times \frac{R2}{R1 + R2} \quad (2.2)$$

Dengan memasukkan nilai yang diinginkan, dengan nilai resistor pertama 1k, didapatkan nilai resistor kedua melalui rumus (2.2) yaitu:

$$\begin{aligned} \frac{3,3}{5} &= \frac{R2}{1000 + R2} \\ 0,66 &= \frac{R2}{1000 + R2} \\ 0,66(1000 + R2) &= R2 \\ 660 &= 0,34 R2 \\ 1941 &= R2 \end{aligned}$$

Sehingga digunakan nilai resistor ke dua yang mendekati yaitu 2,2K. Rangkaian kemudian dihubungkan seperti gambar 7 pada lampiran.

## 2.12 Bekerja dengan Putty dan VNC

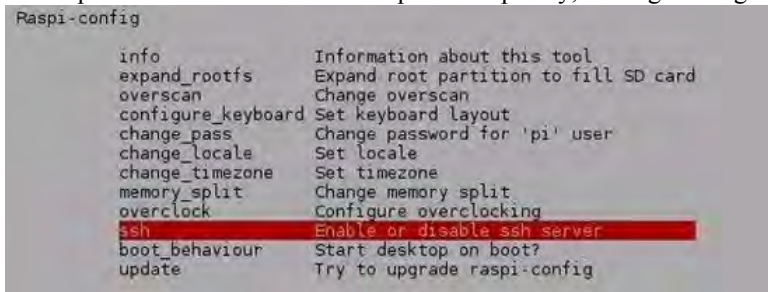
Untuk memudahkan penggunaan raspberry pi tanpa keyboard dan mouse, terdapat pilihan untuk memulai Raspberry PI atau RPi hanya dengan komponen PC/laptop sebagai *peripheral*. Pada umumnya tidak

nyaman untuk bekerja secara langsung pada Raspberry Pi karena dibutuhkan tambahan keyboard dan mouse sehingga masih terkesan kurang *portable*.

VNC (*Virtual Network Connection*) adalah sistem desktop sharing grafis yang memungkinkan pengguna untuk kontrol jarak jauh antarmuka desktop satu komputer dari yang lain. Ini mentransmisikan keyboard dan mouse peristiwa dari controller, dan menerima update ke layar melalui jaringan dari host remote.

Kemudahan yang diberikan adalah tampilan desktop dari Raspberry Pi dalam jendela pada komputer. Kontrol dilakukan sehingga pengguna sedang bekerja pada Raspberry Pi sendiri. Beberapa hal yang harus dilakukan:

Langkah 1: Memulai dan menghubungkan ke SSH Server Arahkan ke ssh dan tekan enter. Ketika diminta tentang server SSH, pilih Enable dan tekan enter lagi. Anda akan kembali ke panel Raspi-config; arahkan ke Finish dan tekan enter. Login dan password SSH default adalah pi dan raspberry, masing-masing.



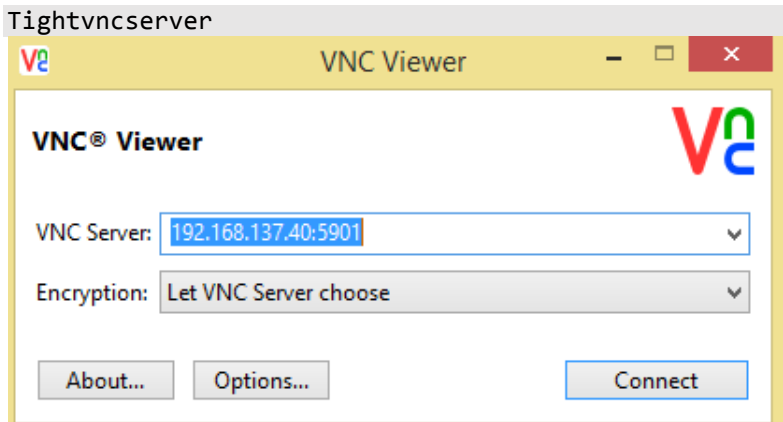
Gambar 2.18 Tampilan Raspi-config untuk meng-enable SSH server

Langkah 3: Setelah SSH diaktifkan, login, dan pengguna mengetahui alamat IP dari mesin (`Sudo ifconfig`), saatnya untuk menghubungkan melalui SSH dan menguji itu. Untuk melakukannya dari Linux dan OS X Anda bisa menggunakan perintah `ssh` di terminal. Pengguna Windows, bagaimanapun, akan membutuhkan klien SSH seperti Putty. Pada interface Sesi utama, cukup ketik alamat IP dari Pi dan pilih SSH dibawahnya:

Pada RPI (menggunakan monitor atau melalui SSH), instal paket TightVNC.

```
sudo apt-get install tightvncserver
```

Langkah 3: Selanjutnya, jalankan TightVNC Server yang akan meminta pengguna untuk memasukkan password dan opsional password:



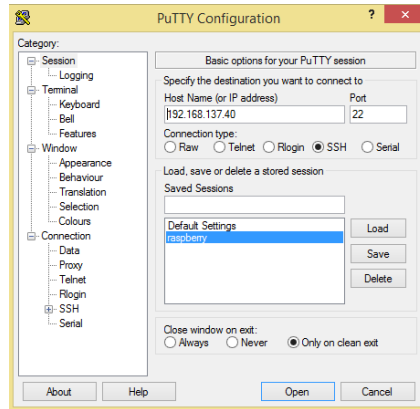
*Gambar 2.19 Konfigurasi awal VNC viewer*

Langkah 3: Sebelum dapat terhubung ke Pi menggunakan VNC, harus terlebih dahulu memulai VNC server. SSH menyediakan mekanisme ini. Untuk mulai dengan, kita mengambil Client SSH kami - Putty. Setelah Anda download, hanya memulai - itu adalah exe mandiri dan ditampilkan pada gambar 2.20

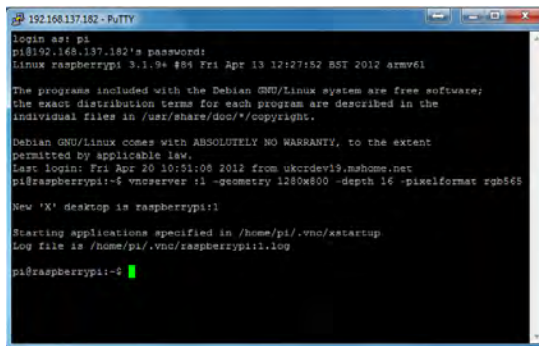
Langkah 4. Setelah melakukan setting IP pada Putty akan masuk halaman untuk login sesuai username dan password default. Jadi hal penting dari perintah ini adalah "1". Ini mendefinisikan port bahwa proses VNC akan dimulai pada RPi sehingga dapat membuat apa pun sesuai kebutuhan.

**vncserver :1**

Jika pengaturan telah dilakukan, maka koneksi VNC viewer dapat memunculkan halaman utama berupa jendela *raspbian* yang telah terpasang. Hal ini memungkinkan penggunaan raspberry tidak lagi terbatas dengan adanya USB keyboard dan USB mouse melalui koneksi laptop atau PC menggunakan kabel LAN (RJ45) melainkan melakukan penggunaan langsung menggunakan piranti laptop atau PC sebagai tampilan antarmuka dengan akses dan pemrosesan yang tetap dijalankan berdasarkan kinerja raspberry pi.



Gambar 2.21 Konfigurasi Putty

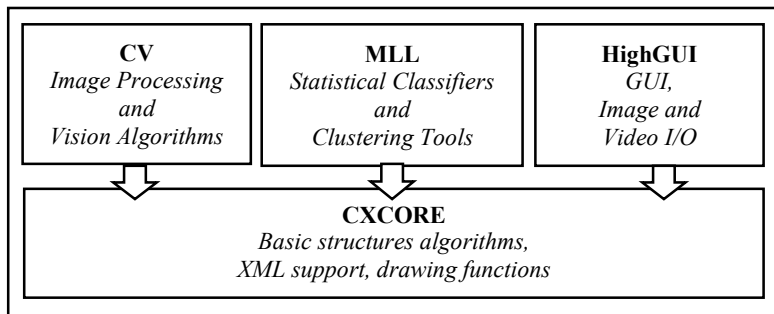


Gambar 2.22 Tampilan halaman login Putty

## 2.13 Memulai Image Processing dengan OpenCV

### 2.13.1 Dasar OpenCV dan bahasa C++

OpenCV disusun menjadi lima komponen utama [11], empat di antaranya ditunjukkan pada Gambar 2.21 dibawah. komponen openCV berisi pengolahan gambar dasar dan algoritma *machine vision* pada tingkat yang lebih tinggi. ML adalah library mesin, yang mencakup banyak pengklasifikasi dan alat pengelompokan statistik. HighGUI berisi I/O dan fungsi untuk menyimpan dan memuat video beserta gambar, dan CXCore berisi struktur data dasar dan konten.



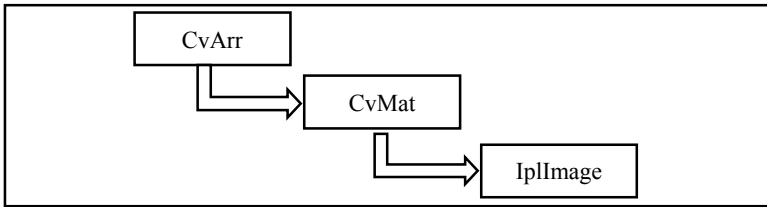
*Gambar 2.23 Struktur dasar OpenCV*

OpenCV memiliki beberapa tipe data primitif. Tipe data ini tidak primitif dari sudut pandang C, tetapi data tersebut merupakan struktur sederhana. Pengguna dapat memeriksa rincian struktur yang dijelaskan dalam file header `cxtypes.h`, yang ada pada “`../OpenCV/cxcore`”, suatu direktori yang ada pada instalasi OpenCV. Hal yang paling sederhana dari jenis ini adalah `CvPoint`. `CvPoint` adalah struktur sederhana dengan dua bilangan bulat anggota, `x` dan `y`. `CvPoint` dibagi menjadi dua; `CvPoint2D32f` dan `CvPoint3D32f`, yang mana keduanya berupa angka floating-point. `CvSize` sama dengan `CvPoint` dengan anggota berupa lebar dan tinggi. Jika pengguna menginginkan angka floating-point, dilakukan dengan menggunakan `CvSize` berupa `CvSize2D32f`. `CvRect` adalah anak lain dari `CvPoint` dan `CvSize`; dengan berisi empat anggota: `x`, `y`, lebar, dan tinggi. Bagian terakhir adalah `CvScalar`, yang merupakan satu set dari empat nomor presisi ganda. `CvScalar` yang sering digunakan untuk mewakili satu, dua, atau tiga bilangan real. `CvScalar` memiliki anggota tunggal “`val`”, yang merupakan pointer ke array yang berisi angka floating-point empat double-presisi. Semua tipe data ini memiliki metode konstruktor dengan nama-nama seperti `cvSize()`.

*Tabel 2.3 Struktur untuk point, size, rect, dan scalar*

Struktur	Isi	Representasi
<code>CvPoint</code>	int <code>x</code> , <code>y</code>	Point in image
<code>CvPoint2D32f</code>	float <code>x</code> , <code>y</code>	Points in $\Re^2$
<code>CvPoint3D32f</code>	float <code>x</code> , <code>y</code> , <code>z</code>	Points in $\Re^3$
<code>CvRect</code>	int <code>x</code> , <code>y</code> , width, height	Portion of image
<code>CvScalar</code>	double <code>val[4]</code>	RGBA value
<code>CvSize</code>	int width, height	Size of image

Tabel 3 menunjukkan kelas atau struktur hirarki tiga jenis gambar. Bila menggunakan OpenCV, pengguna sering kali menghadapi tipe data *IplImage*. *IplImage* adalah struktur dasar yang digunakan untuk mengkodekan apa yang biasanya kita sebut sebagai "gambar". Gambar-gambar ini bisa berupa *grayscale*, warna, empat-kanal (RGB + *alpha*), dan masing-masing saluran dapat berisi salah satu dari beberapa jenis jenis data *floating*. Oleh karena itu, jenis ini lebih umum daripada tiga-channel 8-bit RGB. OpenCV menyediakan kesempatan yang luas dari terhadap gambar-gambar ini, termasuk sistem untuk mengubah ukuran gambar, ekstrak saluran individu, memperoleh nilai terbesar atau terkecil dari saluran tertentu, menambahkan dua gambar, ambang batas (*threshold*) gambar, dan sebagainya.



Gambar 2.24 Meskipun OpenCV diimplementasikan dalam C, struktur yang digunakan dalam OpenCV berorientasi pada objek desain sehingga, *IplImage* berasal dari *CvMat*, yang berasal dari *CvArr*

### 2.13.2 Menampilkan gambar bentuk

Menampilkan suatu bentuk gambar adalah suatu hal yang penting agar pemrosesan dapat dilakukan dengan hasil yang baik dari suatu objek. Untuk itu, OpenCV menyediakan fungsi yang akan memungkinkan kita untuk membuat garis, kotak, lingkaran, dan sejenisnya.

#### 2.13.2.1 Garis

Menggambar garis menggunakan algoritma Bresenham. [Bresenham65]:

```

void cvLine(
    CvArr* array,
    CvPoint pt1,
    CvPoint pt2,
    CvScalar color,
    int thickness = 1,
    int connectivity = 8
  )
  
```

```
);
```

Argumen pertama adalah `cvLine ()`, yang merupakan `CvArr` biasa, dimana dalam konteks ini biasanya berarti gambar pointer *IplImage*. Dua argumen berikutnya adalah `CvPoints`. `CvPoint` adalah struktur sederhana yang hanya berisi anggota bilangan bulat `x` dan `y`. Pengguna bisa membuat `CvPoint` dengan *routine* `CvPoint (int x, int y)`, berupa dua bilangan bulat ke dalam struktur `CvPoint`. Argumen berikutnya, warna, adalah tipe `CvScalar`.

Dua argumen berikutnya adalah jenis yang opsional. Untuk argumen ketebalan merupakan ketebalan garis (dalam pixel), dan untuk argumen konektivitas mengatur mode *anti-aliasing*. Struktur standard adalah "8-connected", yang akan memberikan hasil yang bagus, halus, garis anti-alias. Anda juga dapat mengatur ini ke "4 -connected " line. Pengembangan dari `cvLine ()` adalah `CvRectangle ()`. `CvRectangle ()` menghasilkan bentuk persegi panjang dengan argumen yang sama seperti `cvLine ()` kecuali pada bagian argumen konektivitas. Karena persegi panjang yang dihasilkan selalu berorientasi kepada sisi yang sejajar dengan sumbu `x` dan `y`. Dengan `cvRectangle ()`, hanya diberikan dua poin untuk sudut yang berlawanan dan OpenCV akan menggambar persegi panjang.

```
void cvRectangle(  
    CvArr* array,  
    CvPoint pt1,  
    CvPoint pt2,  
    CvScalar color,  
    int thickness = 1  
);
```

#### 2.13.2.2 Lingkaran dan ellipse

Pada bagian selanjutnya metode untuk menggambar lingkaran, memiliki argumen yang hampir sama.

```
void cvCircle (  
    CvArr* array,  
    CvPoint center,  
    int radius,  
    CvScalar color,  
    int thickness = 1,  
    int connectivity = 8  
);
```



Untuk lingkaran, persegi panjang, dan semua bentuk tertutup lainnya, argumen ketebalan juga dapat diatur melalui CV\_FILL.

Argumen untuk menampilkan gambar elipse:

```
void cvEllipse(  
    CvArr* img,  
    CvPoint center,  
    CvSize axes,  
    double angle,  
    double start_angle,  
    double end_angle,  
    CvScalar color,  
    int thickness = 1,  
    int line_type = 8  
);
```

Hal yang membedakan adalah adanya argumen sumbu yang merupakan jenis CvSize. Fungsi CvSize sangat banyak seperti CvPoint dan CvScalar. CvScalar hanya berisi anggota lebar dan tinggi. Seperti CvPoint dan CvScalar. Argumen tinggi dan lebar mewakili panjang sumbu *major* dan *minor* elips ini. Sudut adalah sudut (dalam derajat) dari sumbu utama, yang diukur berlawanan dari horisontal (yaitu, dari x-axis). Demikian pula *start\_angle* dan *end\_angle* menunjukkan (dalam satuan derajat) sudut untuk busur untuk memulai dan untuk itu untuk ujungnya.

Cara alternatif untuk menentukan gambar elips adalah dengan menggunakan *bounding box*:

```
void cvEllipseBox(  
    CvArr* img,  
    CvBox2D box,  
    CvScalar color,  
    int thickness = 1,  
    int line_type = 8,  
    int shift = 0  
);
```

### 2.13.2.3 Teks

Salah satu bentuk terakhir dari gambar adalah menampilkan teks. OpenCV memiliki satu kebiasaan utama, disebut cvPutText ():

```
void cvPutText(  
    CvArr* img,  
    const char* text,  
    CvPoint origin,
```

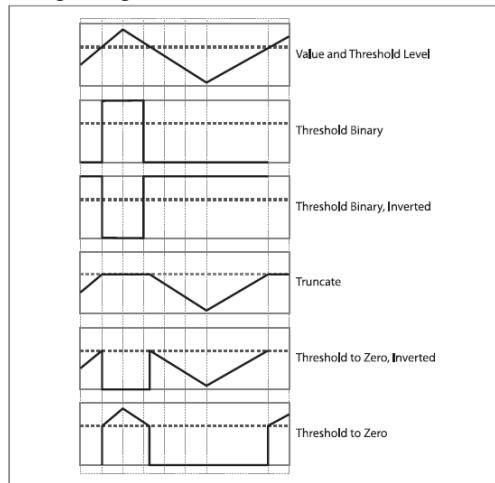
```
const CvFont* font,
CvScalar color
);
```

### 2.13.3 Smoothing

Dalam pengolahan citra, pengaburan, , juga disebut *blurring* atau *smoothing*, adalah operasi pengolahan citra sederhana [11] dan sering digunakan. Ada banyak alasan untuk menghaluskan, tetapi biasanya dilakukan untuk mengurangi *noise* atau artefak kamera. Smoothing juga penting ketika kita ingin mengurangi resolusi dari suatu gambar.

### 2.13.4 Ambang Batas

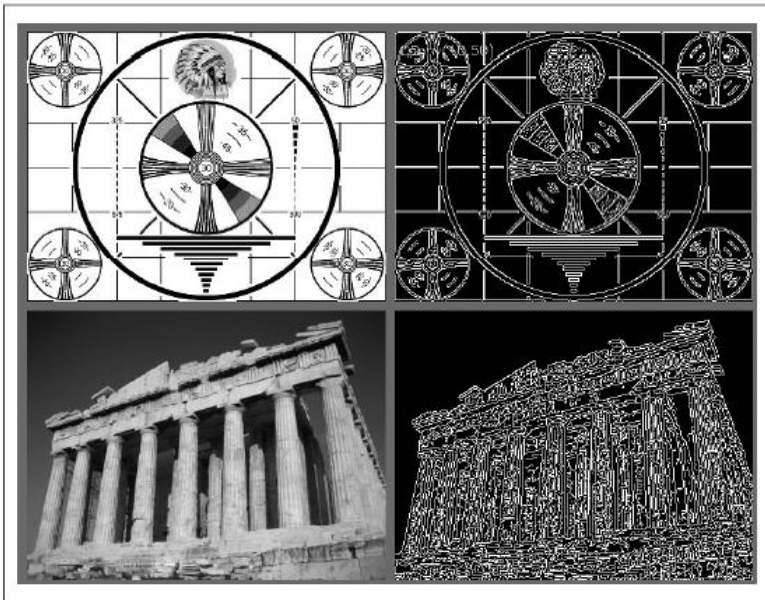
Dalam pengolahan citra dan menghasilkan keluaran yang sesuai dengan keinginan pengguna, juga diperlukan keputusan tentang kategori khusus berupa piksel dalam gambar dengan menetapkan *threshold* atau ambang batas [4]. Ambang batas akan menggunakan suatu nilai dalam range dan akan membuang nilai yang berada diluar range tersebut, baik diatas maupun dibawah nilai yang ditentukan. Fungsi `cvThreshold()` menyelesaikan tugas-tugas ini.



Gambar 2.25 Hasil dari berbagai jenis ambang batas di `cvThreshold()`. Garis horizontal melalui setiap grafik merupakan ambang batas tertentu diterapkan pada grafik atas dan efeknya untuk masing-masing lima jenis operasi ambang bawah

### 2.13.5 Deteksi Tepi Canny

Metode untuk menemukan tepi disempurnakan oleh J. Canny pada tahun 1986 menjadi apa yang sekarang biasa disebut sebagai *Canny edge detection* (detektor tepi Canny) [9]. Salah satu perbedaan antara algoritma Canny dan algoritma lain, misalnya algoritma *Laplace* berbasis, yaitu dalam algoritma Canny, turunan pertama dihitung di  $x$  dan  $y$  dan kemudian digabungkan menjadi empat arah derivatif. Poin dari arah derivatif tersebut adalah *local maxima* dan menjadi cikal bakal untuk membentuk ujung dari sudut.



Gambar 2.26 Hasil Canny edge detection untuk dua gambar ketika diberi pengaturan batas tinggi dan rendah

### 2.13.6 Lingkaran Hough

Lingkaran Hough (*Hough Circle*) [11] adalah salah satu metode dalam pengolahan citra untuk mendapatkan suatu bentuk lingkaran dari objek gambar. Deteksi warna biasanya dilakukan sehingga bentuk lingkaran dapat terdeteksi letak, dan nilai ukurannya.

### 2.13.7 Kontur

Walaupun algorithma seperti deteksi tepi Canny dapat digunakan untuk mendapatkan *edge* dari pixel- pixel yang memisahkan segmentasi yang berbeda dalam suatu gambar, metode itu tidak memberi tahu apapun tentang *edge* atau sudut-sudut tersebut sebagai *entities* dalam gambar tersebut [11]. Langkah selanjutnya adalah menyusun sudut piksel kedalam bentuk kontur (*contours*). Salah satu yang juga diperukan untuk proses menghasilkan kontur adalah dengan *memory storage* atau gudang penyimpanan memori, dimana penyimpanan memori memungkinkan OpenCV untuk membentuk objek baru yang dinamis. Kemudian dilanjutkan ke tata urutan sehingga akan memunculkan *contours*.

#### 2.13.7.1 Memory Storage

OpenCV menggunakan gudang penyimpanan memori untuk mengatasi alokasi memori untuk membentuk objek yang dinamis. Penyimpanan memory ini dihubungkan dengan suatu alur dari blok memori yang mengijinkan alokasi secara cepat dan mengatur ulang alokasi dari dari blok memori secara kontinyu. Ditampilkan dalam 4 buah *routines*:

```
CvMemStorage* cvCreateMemStorage(  
    int block_size = 0  
);  
  
void cvReleaseMemStorage(  
    CvMemStorage** storage  
);  
  
void cvClearMemStorage(  
    CvMemStorage* storage  
);  
  
void* cvMemStorageAlloc(  
    CvMemStorage* storage,  
    size_t size  
);
```

Untuk membuat sebuah penyimpanan memori digunakan fungsi “cvCreateMemStorage()” fungsi ini menggunakan argumen ukuran blok yang memberikan ukuran dari memory didalam sistem penyimpanan. Nilai 0 akan menetapkan nilai *default* blok (64kB).

“cvReleaseMemStorage()” merupakan fungsi untuk meletakkan pointer ke dalam memori penyimpanan dan menata ulang alokasi memori yang dibutuhkan untuk data gambar, matriks, dan struktur lain.

Dengan memanggil “cvClearMemStorage()”, dapat dilakukan pembersihan data.

#### 2.13.7.2 Sequences

Salah satu jenis objek yang dapat disimpan kedalam memori penyimpanan sekuensial. OpenCV dapat membuat sekuensial dari beberapa jenis obyek. Penyusunan dari Struktur *CvSeq*:

```
typedef struct CvSeq {
    int      flags;           // miscellaneous flags
    int      header_size;    // size of sequence header
    CvSeq*   h_prev;         // previous sequence
    CvSeq*   h_next;         // next sequence
    CvSeq*   v_prev;         // 2nd previous sequence
    CvSeq*   v_next;         // 2nd next sequence
    int      total;          // total number of elements
    int      elem_size;      // size of sequence element in
byte
    char*    block_max;      // maximal bound of the last
block
    char*    ptr;            // current write pointer
    int      delta_elems;    // how many elements allocated
                                // when the sequence grows
    CvMemStorage* storage;   // where the sequence is stored
    CvSeqBlock* free_blocks; // free blocks list
    CvSeqBlock* first;       // pointer to the first sequence
block
}
```

#### 2.13.7.3 Menemukan Contour Bounding Box

Pengertian dari *contour* adalah kumpulan dari titik yang merepresentasikan kurva didalam suatu gambar yang berbeda tergantung pengaturan. Terdapat banyak cara untuk merepresentasikan kurva tersebut. Contours direpresentasikan dengan mengkodekan informasi tentang lokasi titik selanjutnya pada kurva di sf a. fww

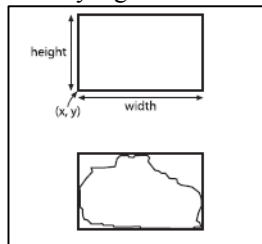
Subroutine *cvContourPerimeter()* akan mengambil kontur dan panjangnya. Bahkan, fungsi ini sebenarnya adalah fungsi makro untuk *cvArcLength()*. Fungsi yang memiliki relasi dengan fungsi

*cvArcLength()* adalah *cvContourArea()*, yang mana fungsi ini menghitung area yang ada dalam kontur.

Panjang dan luas area merupakan karakteristik dari suatu kontur. Pengembangannya adalah berupa *bounding box* atau *bounding circle/ellipse*. Cara untuk membentuknya:

```
CvRect cvBoundingRect(
    CvArr* points,
    int update = 0
);
CvBox2D cvMinAreaRect2(
    const CvArr* points,
    CvMemStorage* storage = NULL
);
```

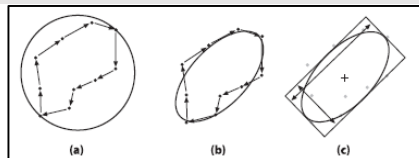
Cara termudah adalah dengan *cvBoundingRect()*. Fungsi ini apat menampilkan persegi yang merepresentasikan garis yang terdiri dari bagian horizontal dan vertikal yang memutar objek.



Gambar 2.27 Persegi sebagai bounding rect

Sedangkan untuk *cvMinEnclosingCircle()* bekerja hampir sama dengan fungsi bounding box, dengan fleksibilitas untuk memberi tanda pada sekuense atau *array* dari titik dua dimensi.

```
CvBox2D cvFitEllipse2(
    const CvArr* points
);
```



Gambar 2.28 Kontur berupa sepuluh poin dengan melampirkan lingkaran minimal (a) dengan ketepatan ellipsoid (b); kotak (c) digunakan oleh OpenCV untuk mewakili ellipsoid

**Halaman ini sengaja dikosongkan**

## **BAB 3**

### **PERANCANGAN SISTEM**

Metodologi Pelaksanaan guna mencapai tujuan yang telah dirumuskan, terdapat 9 tahap dimulai dari analisa masalah pada mitra, perancangan dan pembuatan alat, hingga alat dalam kondisi siap dan dapat diterapkembangkan. Flowchart metodologi program yaitu:



Gambar 3.1 Flowchart pelaksanaan kegiatan

#### **3.1 Observasi dan Analisa Masalah**

Dalam perancangan implementasi alat harus dilakukan pengamatan lokasi yang akan dituju terkait dengan masalah yang diadapi, sehingga perancangan akan tepat guna dan pemakaiannya efisien. Dalam hal ini masalah yang ada adalah tidak efektifnya pengukuran diameter kayu untuk mendapatkan nilai isi kayu bundar secara konvensional menggunakan meteran. Hal tersebut diakibatkan



oleh adanya galat atau *error*. Pada tahap ini dilakukan pengamatan terhadap permasalahan, situasi dan kondisi yang terjadi pada proses pengukuran di industri kayu, permasalahan selanjutnya akan dianalisa dan nantinya akan dipadukan dengan teknologi pada alat.



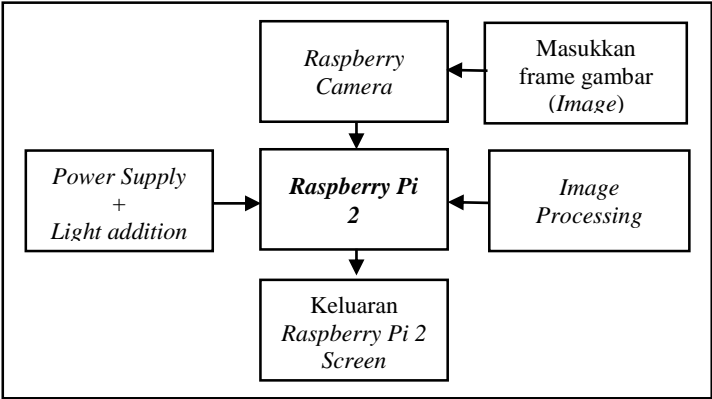
Gambar 3.2 Industri pengolahan kayu PT. Indofurnitama Raya, Pasuruan

### 3.2 Studi Literatur

Studi literatur berisi serangkaian kegiatan dan pengkajian sumber-sumber yang relevan dan terpercaya dalam pengumpulan materi serta menjadi acuan dalam penulisan ini. Kami menggunakan banyak literatur berbeda agar dapat menghasilkan informasi yang lengkap.

### 3.3 Perancangan Sistem

Perancangan sistem dila diawali dengan pemrograman visual dengan menggunakan *software OpenCV 2.4.9* dan *CodeBlocks* (keduanya open source), lalu setelah itu pengintegrasian antara program dengan hardware yakni dengan Raspberry pi, dengan mengandalkan kinerja *machine vision* maka hasil pengukuran yang presisi akan didapat dan akan ditampilkan pada layar lcd (*screen*).



Gambar 3.3 Rancangan sistem utama

### 3.4 Pembuatan Alat

Perancangan alat ini bertujuan untuk mendapatkan desain dan mekanisme yang optimal dengan memperhatikan data yang telah didapat dari studi literature dan observasi. Adapun rancangan kayu meter sebagaimana ditunjukkan pada lampiran gambaran teknologi yang akan diterapkembangkan, dengan desain pada gambar 6. Dengan ditunjang dengan desain ukuran seminim mungkin, alat kayu meter dapat diaplikasikan karena mudah dibawa. Gambar 3.2 menunjukan perancangan *hardware* sistem.

*Gambar tidak untuk dipublikasikan*

Gambar 3.4 Gambaran alat yang diterapkembangkan

Keterangan :

1. Kamera

Kamera berfungsi sebagai perangkat pembaca agar didapatkan visualisasi kayu sehingga dapat diproses.

2. Layar

Untuk menampilkan data yang telah diproses. Layar dapat menggunakan monitor, touchscreen ataupun laptop melalui HDMI.

3. Raspberry Pi 2

Raspberry Pi menjadi pusat control dan proses yang terhubung pada semua device. Dengan tambahan aplikasi

4. Micro SD

Micro SD berfungsi sebagai media penyimpanan.

5. Power Supply

Dibutuhkan catu daya untuk menghidupkan device dengan spesifikasi minimal 5 DC volt, 700 mA Menggunakan

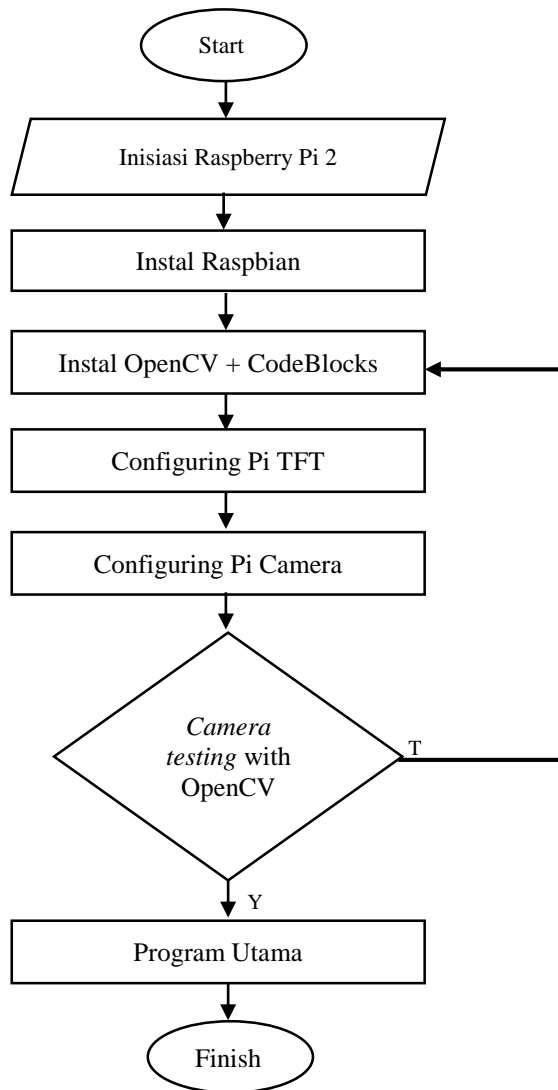
Gambar 3.5 Keterangan gambaran alat yang diterapkembangkan

### 3.5 Perancangan Program dan Pengujian


Pengujian dilakukan di lokasi industri di yaitu dengan mengukur secara langsung objek kayu bundar menggunakan alat kayu meter dan akan dievaluasi hasil pengukuran apakah sudah sesuai dengan hasil sesungguhnya.

*Gambar tidak untuk dipublikasikan*

Gambar 3.6 Visualisasi uji coba devais saat melakukan pengukuran terhadap kayu



Gambar 3.7 Flowchart alur bagian *hardware* sistem



*Gambar tidak untuk dipublikasikan*

Gambar 3.8 Flowchart algoritma *software* sistem

### **3.6 Pengumpulan Data dan Evaluasi**

Data yang diperoleh dari hasil pengukuran yang dilakukan akan segera dievaluasi mengenai kepresisian data yang dihasilkan. Perbaikan sistem terhadap tingkat kepresisian yang dihasilkan.

### **3.7 Penyusunan Laporan**

Tahap akhir dari rangkaian ini adalah penyusunan dan pembuatan sebagai laporan. Selain itu dapat digunakan sebagai bahan referensi dan dokumentasi penyelesaian sistem.

*Halaman ini sengaja dikosongkan*

## **BAB 4**

### **PENGUJIAN DAN ANALISIS SISTEM**

Setelah mengetahui kesiapan fungsi dari bagian hardware pengujian dilanjutkan ke tahap algoritma program melalui dua bagian metode utama, yaitu pengujian terhadap video kayu dan pengujian terhadap *prototype* kayu dengan berbagai ukuran. *Sample* untuk kayu digunakan beberapa ukuran yang menjadi ukuran umum pada industri kayu PT Indofurnitama Raya, Pasuruan.



*Gambar 4.1 Frame kayu yang akan diujicoba dengan berbagai ukuran, 19 cm (tampak kiri) dan 18 cm (tampak kanan)*

#### **4.1 Pengujian Fungsi *Hardware***

Hardware yang digunakan adalah hardware untuk menjalankan fungsi standard *Raspberry Pi* dan dikembangkan untuk dapat menjalankan fungsi *image processing* menggunakan OpenCV.

##### **4.1.1 Menampilkan Raspbian**

Operating Sistem yang digunakan adalah *Raspbian wheezy* versi 2015-02-16. Melalui kabel HDMI, raspberry dikoneksikan ke layar monitor sehingga OS dapat ditampilkan.

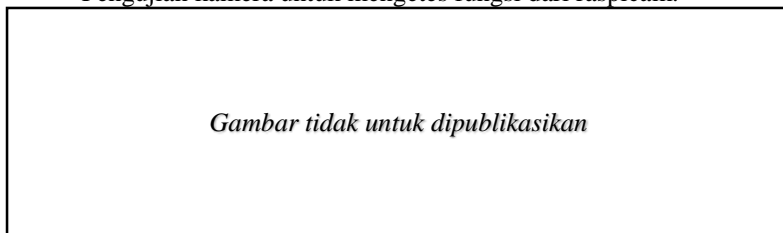
##### **4.1.2 Konfigurasi Pi TFT**

*Pi TFT* sebagai monitor utama untuk menunjang tingkat portabel alat dengan kemampuan untuk menampilkan OS sistem dan menjadi *user interface*.



### 4.1.3 Tes Kamera Raspberry Pi (*RaspiCam*)

## Pengujian kamera untuk mengetes fungsi dari raspicam.



*Gambar 4.2 Pengujian kamera raspberry pi (tampak kiri) dan hasil yang ditampilkan kamera (tampak kanan)*

#### 4.1.4 Konfigurasi *OpenCV* dan *CodeBlocks* pada *Raspberry Pi*

#### 4.1.4.1 Pengujian Pengujian Library dan Header OpenCV Melalui Proses Build and Run

Proses *build and run* dilakukan untuk menguji fungsi library dan header yang telah diinstall. Keberhasilan dapat dilihat dengan tidak adanya status *error* pada *syntax* program yang dibuat.



*Gambar 4.3 Proses build dan run pada CodeBlocks berjalan dengan baik*

## 4.2 Pengujian Algoritma

Tahap pengujian pengukuran dilakukan untuk mengetahui metode yang paling tepat dengan akurasi yang paling tinggi dalam proses mendapatkan nilai diameter minimum dalam penentuan grade dari suatu kayu. Berdasarkan studi pustaka dari berbagai referensi yang telah dirancang pada bagian metodologi penelitian, beberapa algoritma yang dilakukan untuk menyelesaikan masalah tersebut adalah dengan *hough circle*, *contours*, dan *bounding box*. Dari berbagai algoritma tersebut akan dicari metode yang memiliki tingkat akurasi tertinggi dalam menentukan nilai diameter terkecil dari kayu. Langkah-langkah dilakukan dalam pengujian terhadap bahan uji berupa 3 buah prototype, dan 3 buah sampel nyata kayu industri yang diperoleh dari Indofurnitama Raya, Pasuruan.

*Tabel 4.1 Rekapitulasi Pengujian Hardware Raspberry Pi 2*

*Tabel tidak untuk dipublikasikan*

#### **4.2.1 Pengamatan Objek Uji Coba**

Pengujian dilakukan terhadap beberapa jenis bahan uji. Perhitungan awal melalui pengamatan dilakukan untuk mengetahui nilai sebenarnya dari bahan uji untuk dibandingkan dengan nilai dari alat.

- **Prototype A**

Nilai terukur: 18.1 cm

Nilai grade : 18

\*catatan : garis merah menunjukkan  
diameter terkecil kayu

*Gambar 4.4 Gambar prototype A dengan grade 18*

- **Prototype B**

Nilai terukur: 13.8 cm

Nilai grade : 14

\*catatan : garis merah menunjukkan  
diameter terkecil kayu

*Gambar 4.5 Gambar prototype B dengan grade 14*

- Prototype C

Nilai terukur: 18.8 cm

Nilai grade : 19

\*catatan : garis merah menunjukan  
diameter terkecil kayu

*Gambar 4.6 Gambar prototype C dengan grade 19*

- Kayu A

Nilai grade : 17

*Gambar 4.7 Gambar nyata kayu yang diuji dengan grade 19*

- Kayu B

Nilai grade : 18

*Gambar 4.8 Gambar nyata kayu B yang diuji dengan grade 18*

- Kayu C

Nilai grade : 19

*Gambar 4.9 Gambar nyata kayu B yang diuji dengan grade 17*

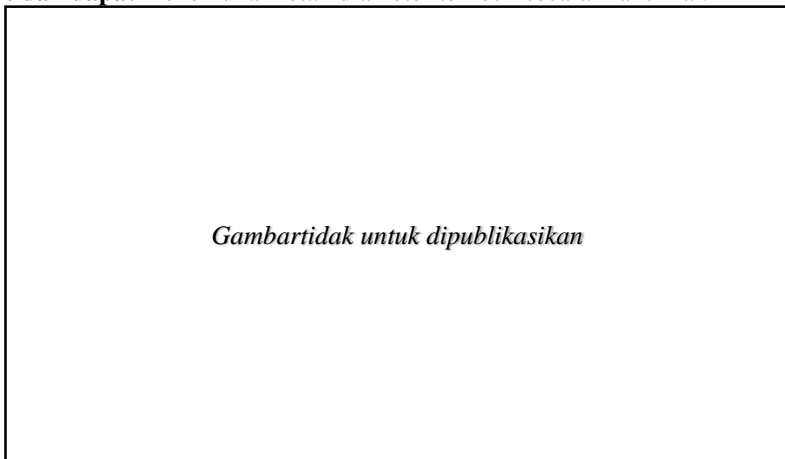
#### 4.2.2 Kalibrasi Jarak Awal 30 cm

Kalibrasi dilakukan dengan menggunakan jarak 30 cm, dengan obyek berdiameter terkecil 17,2 cm, sehingga diperoleh jarak aktual dengan nilai perbandingan gambar nyata dan gambar dalam pixel adalah 1: 4.7.

#### 4.2.3 Pengujian Algoritma Hough Circle

Pengujian untuk menentukan nilai diameter terkecil dari kayu dilakukan dengan metode *hough circle* yang berbasis warna. Pengaturan warna diberikan dengan nilai RGB atas (64,119,61) dan RGB bawah (159, 191, 81) pada *prototype* C dengan nilai pengukuran 18,8 (grade 19). Pengujian yang dilakukan menunjukkan algoritma tidak cukup tangguh untuk menemukan bentuk lingkaran karena nilai berubah-ubah akibat pengaruh kepekaan terhadap intensitas cahaya yang berimbas pada proses deteksi warna.

Nilai deteksi lingkaran berubah dengan range error sampai dengan 9.14 % dengan cuplikan satu detik frame bernilai masukan antara 18.374746 s.d. 20.529987 pada posisi tetap dan berjarak 30 cm. Sehingga sistem **tidak efektif** digunakan untuk pengukuran. Sistem juga **tidak dapat** menemukan letak diameter terkecil secara maksimal.



*Gambar 4.10 Tampilan pegujian program menggunakan algoritma hough circle. Nilai yang dihasilkan memiliki error s.d. 11% (terlalu besar)*

#### 4.2.4 Pengujian Deteksi Kontur dan *Bounding Box*

Sistem mampu menemukan objek yang berbentuk bulat sempurna, ataupun yang tidak berbentuk bulat sempurna. Dengan kontur dan *bounding box*, sistem dapat mendeteksi bagian dari permukaan kayu yang akan diukur dan secara akurat dapat menentukan letak diameter terkecil beserta nilainya.

#### 4.2.5 Pengujian Sistem pada *Prototype*

Pengujian dilakukan terhadap 6 jenis bahan uji, yaitu prototype A, prototype B, prototype C, Kayu A, Kayu B, Kayu C yang telah didefinisikan pada subbab 4.2.1. *Prototype* kayu dibuat dengan bentuk kontur yang tidak bulat sempurna untuk mendeteksi kehandalan sistem. Hasil pengujian pada *prototype* ditunjukkan pada tabel 4.2.

Pengujian menunjukan metode kontur dan *bounding box* memiliki tingkat akurasi mencapai 97%. Tingkat tersebut memiliki eror yang kecil dan pengujian sistem dapat dilanjutkan ke tahap selanjutnya.

*Tabel tidak untuk dipublikasikan*

Tabel 4.2 Hasil Pengujian kehandalan alat pada prototype



*Gambar 4.11 Screenshot pengujian terhadap prototype A (grade 18)*



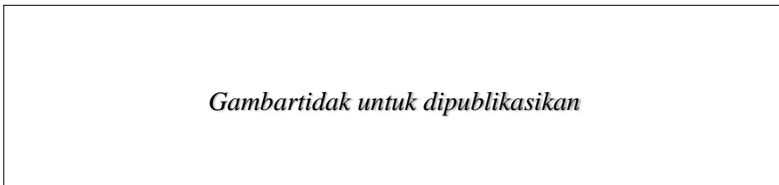
*Gambar 4.12 Screenshot pengujian terhadap prototype B (grade 14)*



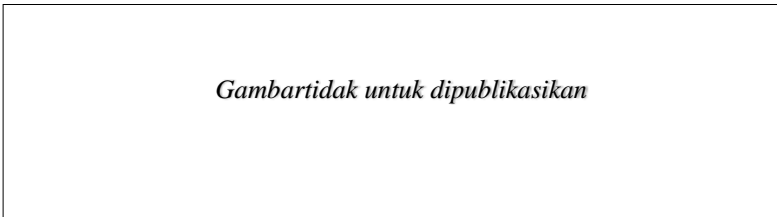
*Gambar 4.13 Screenshot pengujian terhadap prototype C (grade 19)*

#### **4.2.6 Menemukan Nilai Diameter Terkecil**

Sistem *bounding box* secara akurat dapat menemukan letak diameter terkecil pada masing- masing bahan uji. Hal ini ditunjukkan pada gambar dibawah:



*Gambar 4.14 Sistem menentukan diameter terkecil pada prototype (Status: Akurat)*



*Gambar 4.15 Sistem menentukan diameter terkecil pada prototype (Status: Akurat)*

*Gambartidak untuk dipublikasikan*

*Gambar 4.16 Sistem menentukan diameter terkecil pada prototype (Status: Akurat)*

### **4.3 Pengujian Keandalan Alat Menggunakan Ultrasonik SR04**

#### **4.3.1 Kalibrasi Sistem dengan Sensor Ultrasonik**

Kalibrasi dilakukan untuk mendapatkan nilai terukur dari ultrasonik dan nilai yang didapatkan oleh Raspberry Pi camera. Perhitungan jarak (Range\_cm) dilakukan dengan mendapatkan waktu tempuh (time\_diff) dari selisih waktu yang ditempuh dari jarak pengiriman sinyal TRIG dan penerimaan sinyal ECHO dengan kondisi suhu lingkungan ideal dan cepat rambat gelombang di udara 343 m/detik. Setelah didapatkan nilai jarak, dilakukan kalibrasi dengan hasil yang ditangkap oleh kamera kamera Raspberry Pi. Pengujian dilakukan terhadap suatu kayu bundar dengan grade 15, yang didapatkan nilai pengukurannya dari hasil proses kamera.

*Gambartidak untuk dipublikasikan*

*Gambar 4.17 Objek Pengujian sistem pada kayu grade 15 (Gambar kanan) dan nilai sebenarnya melalui pengukuran konvensional. Diameter terkecil ditunjukan oleh garis oranye dengan nilai 15,07 cm*



Setelah mendapatkan nilai sebenarnya dari kayu grade 15, dilakukan pengukuran dengan pengujian pengaruh jarak terhadap nilai untuk didapatkan rumus yang paling ideal.

*Gambartidak untuk dipublikasikan*

*Gambar 4.18 Kalibrasi ultrasonik SR04 terhadap kamera dengan jarak 40cm, 30 cm, dan 20 cm*

*Tabel 4.3 Tabel hasil kalbrasi ultrasonik dan kamera pada objek grade 15*

No.	Jarak	Nilai Diameter Terkecil	Nilai Diameter Terbesar
1.	20	106,3	117,8
2.	30	70,71	77,31
3.	40	52,0	58,1

Dari hasil pada tabel diperoleh nilai perbesaran pixel akibat pengaruh jarak dengan masing masing untuk 20 cm terjadi perbesaran 7,05375 kali, untuk 30 cm terjadi 4,692 kali, dan untuk 40 cm terjadi perbesaran 3,525 kali. Konstanta kalibrasi yang merupakan nilai perkalian antara jarak dan perbesaran, didapatkan nilai 141, sehingga rumus untuk kalibrasi adalah:

$$\text{Nilai akhi} = \frac{\text{Nilai diameter}}{\left( \frac{141}{\text{Jarak terukur}} \right)} \quad (4.1)$$

#### 4.3.2 Pengujian Dinamisasi Jarak dengan Ultrasonik

Pengujian kehandalan alat dilakukan terhadap tiga buah kayu dengan grade 15. Sistem memanfaatkan sensor ultrasonik SR04 sebagai penghitung jarak. Kondisi lingkungan dengan cahaya didalam ruang pada malam hari, sehingga diberikan cahaya tambahan dari LED. Jarak sekitar 20 cm - 30 cm. Alat digerakkan ke tiga posisi untuk memproses tiga buah kayu secara kontinyu dan dilakukan pengambilan data untuk menghitung *error*.




*Gambar 4.19 Objek kayu uji dengan grade 15*

*Tabel 4.4 Hasil pengujian sistem dengan ultrasonik terhadap tiga buah kayu bundar grade 15. Tabel dengan warna dasar merah muda nilainya diabaikan karena nilai tersebut merupakan perpindahan posisi.*

*Tabel tidak untuk dipublikasikan*

*Tabel tidak untuk dipublikasikan*

Berdasarkan tabel 4.4, hasil pengukuran menggunakan tambahan ultrasonik menunjukkan sistem dapat bekerja dengan baik dengan kondisi jarak ideal 20-30 cm dengan catatan sistem dapat menangkap kontur dari kayu yang terukur. Error rata-rata yang didapatkan sekitar 2,118% dengan jangkauan error maksimal mencapai 5,9% sehingga akurasi mencapai 94,1% saat kondisi stabil (bukan perpindahan). Secara umum sistem dapat dilihat melalui grafik pengukuran dengan ultrasonik.



*Gambar tidak untuk dipublikasikan*

*Gambar 4.20 Grafik pengujian ultrasonik terukur tanpa nilai perpindahan.*

#### **4.4 Pengujian Sistem pada Kayu Nyata Industri**

Pengujian dilakukan pada kayu nyata industri melalui sampel yang diambil dari Indofurnitama Raya, Pasuruan. Nilai ditampilkan pada tabel 4.4. Kondisi uji coba dilakukan dalam kondisi di luar ruangan pada siang hari dengan intensitas cahaya dan suhu normal.

Pengujian dilakukan pada tiga buah sampel yang telah diukur menggunakan meteran dan diperoleh nilai grade yaitu grade 17, grade 18, dan grade 19. Nilai diameter terkecil melalui algoritma kontur dan *bounding box* akan ditampilkan melalui tampilan antarmuka pada gambar 4.21 dan 4.22. Melalui tabel, akan ditunjukkan nilai eror minimal dan eror maksimal untuk menunjukkan akurasi sistem dalam kondisi ideal.

*Tabel 4.5 Hasil Pengujian pada Kayu Nyata Industri*

*Tabel tidak untuk dipublikasikan*


*Gambar tidak untuk dipublikasikan*

*Gambar 4.21 Screenshot pengujian terhadap kayu A (grade 17)*



*Gambar tidak untuk dipublikasikan*

*Gambar 4.22 Screenshot pengujian terhadap kayu B (grade 18)*



*Gambar tidak untuk dipublikasikan*

*Gambar 4.23 Screenshot pengujian terhadap kayu C (grade 19)*

#### **4.4.1 Ketahanan Power Supply**

Power Supply yang digunakan adalah berupa power bank “Hame” *good quality* dengan kapasitas arus 11.000 mAh dengan keluaran 5 Volt- 2 Ampere, sehingga dengan daya tersebut, dengan penggunaan 5 Volt- 700mA s.d. 1000mA tanpa *overclock* alat akan dapat digunakan selama 11 jam tanpa berhenti.

#### **4.4.2 Portabilitas Alat**

Salah satu keunggulan yang harus didapatkan dari alat adalah tingkat portabilitas dengan berat yang ideal dengan ukuran yang tidak besar. Spesifikasi dari alat yaitu:

Berat : 500 gr

Dimensi : 90 x 55 x 60 mm

Spesifikasi tersebut menunjukkan alat relatif tidak terlalu berat dan dapat digunakan secara portabel sehingga menunjang portabilitas.

#### **4.4.3 Kemudahan Duplikasi Alat**

Duplikasi alat dengan hanya menggunakan microSD/SD *card*. File yang berupa file *image* tanpa merubah isi program diduplikat penyimpanan.

#### **4.4.4 Potensi Pengguna**

Sistem standarisasi pengukuran kayu yang telah dirancang, dapat dikembangkan dengan potensi pengguna yang cukup luas. Karena terdapat sekitar lima ribu perusahaan kayu yang terdaftar di seluruh Indonesia. Hal ini belum termasuk badan usaha milik masyarakat yang ada di daerah.

Selain itu, instansi yang berwenang untuk melakukan pengukuran kayu yaitu Perhutani dan Kementrian Kehutanan dan Dinas terkait, juga membutuhkan alat ukur kayu untuk melakukan sertifikasi terhadap kayu industri dan melakukan pendataan terhadap kayu industri menjadi suatu dokumen resmi.

#### 4.4.5 Analisa Fungsi Sistem

*Outline* metode ditampilkan oleh gambar 4.24. Pada bagian tersebut dijelaskan keberhasilan alur program yang telah dirancang membentuk nilai diameter minimal dan mendeteksi *contours* dari kayu dengan tingkat ketelitian mencapai 97%. Berbagai proses pengolahan citra ditampilkan oleh gambar 4.25.



*Gambartidak untuk dipublikasikan*

*Gambar 0.1 Outline metode*



Sistem yang diaplikasikan meningkatkan efektifitas pengukuran pada industri kayu. Jika biasanya pengukuran secara manual menggunakan alat bantu penggaris butuh waktu 15- 20 detik untuk mendapatkan nilai terukur, dengan menggunakan alat ini bisa menjadi 5-10 detik atau terjadi peningkatan efisiensi sekitar 50 %. Hal itu akan sangat membantu dengan banyaknya jumlah kayu pada industri.

*Gambartidak untuk dipublikasikan*

*Gambar 4.25 Berbagai tahap pengolahan citra pada sistem (a) Gambar asli (b) Gambar mode grayscale (c) Smoothing (d) Mendapatkan edge dengan threshold (e) Bounding Box (f) Hasil akhir dengan memberikan label nilai diameter terkecil*

## BAB 5

### PENUTUP

#### 5.1 Kesimpulan

Sistem untuk melakukan pengukuran pada kayu bundar diperlukan sebagai standarisasi cara pengukuran kayu dimana cara yang ada saat ini berpotensi merugikan berbagai pihak. Hal ini dikarenakan penggunaan alat bantu penggaris memiliki potensi kesalahan akibat perbedaan persepsi pengukuran dalam menentukan nilai diameter saat menentukan nilai grade suatu kayu. Sehingga diperlukan devais pengukuran untuk mengatasi masalah kesalahan cara pengukuran tersebut. Melalui penelitian dan pembuatan alat pengukur diameter kayu berbasis raspberry pi dengan menggunakan pengolahan citra, dapat diperoleh kesimpulan yaitu:

1. Alat pengukuran berbasis raspberry pi menggunakan metode *contours* dan *bounding box* mampu menemukan diameter terkecil dan memiliki tingkat akurasi mencapai 97% dibandingkan penggunaan metode lain seperti *hough circle* yang akurasinya mencapai 89% namun tidak stabil dalam menghasilkan deteksi berbasis warna. Sehingga sistem yang dibuat layak digunakan sebagai alat standarisasi pengukuran kayu. Dengan menggunakan sensor ultrasonik HC-SR04 untuk meningkatkan fleksibilitas alat dapat mempengaruhi akurasi menjadi 94,1% saat kondisi stabil dengan cahaya yang cukup untuk menangkap kontur. Keandalan untuk menangkap gambar menggunakan devais tambahan ultrasonik HC-SR04 dipengaruhi oleh variabel nilai diameter yang ditangkap kamera dan nilai jarak yang diperoleh sensor ultrasonik. Sehingga variabel *error* berpotensi lebih besar daripada menggunakan jarak yang dibuat tetap.
2. Sistem portabel yang dibuat dapat menemukan letak diameter terkecil untuk memperpendek waktu sehingga sangat membantu proses pengukuran industri kayu.
3. Sistem berbasis *raspberry pi* menunjang untuk dijadikan devais alat bantu pengukuran dikarenakan berat dan ukuran yang relatif ideal untuk digunakan sebagai alat portabel.

## 5.2 Saran

Saran yang dapat diberikan untuk pengembangan alat ke depan antara lain sebagai berikut:

1. Pengembangan algoritma program berbasis pengolahan citra untuk mendapatkan nilai pengukuran yang stabil menuju akurasi 100% dengan tingkat portabilitas tinggi.
2. Pengembangan dinamisasi jarak menggunakan devais ultrasonik, koordinat *gyro* atau alat pengukur jarak lain yang memiliki akurasi yang baik.
3. Sistem pengukuran kayu yang dirancang, mampu menggantikan peran fungsi pengukur konvensional seperti penggaris dan meteran, namun hal ini masih terbatas pada pengukuran terhadap satu atau dua obyek saja. Pengembangan selanjutnya dapat dilakukan dengan memanfaatkan pengolahan citra terhadap tumpukan kayu bundar dalam sekali proses pengolahan, didapatkan nilai masing- masing diameter kayu sekaligus, sehingga akan meningkatkan efektivitas waktu dalam pengukuran dengan catatan tetap memperhatikan kemampuan deteksi obyek, segmentasi obyek, dan akurasi pengukuran.

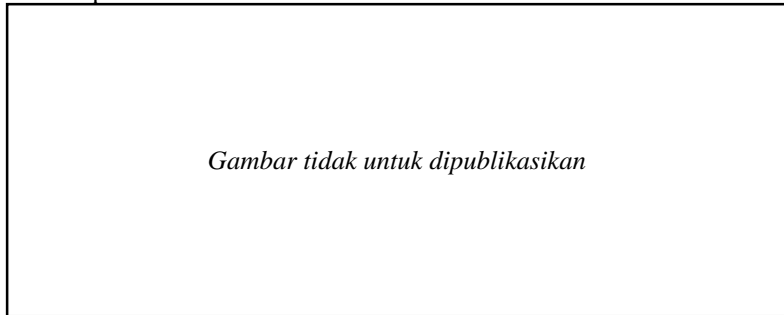
## DAFTAR PUSTAKA

- [1] J.E. Tomaszewski, J. Hipp, M. Tangrea, A. Madabhushi *Machine Vision and Machine Learning in Digital Pathology, Pathobiology of Human Disease*, 2014, hal 3711-3722, New York
- [2] Ha, Jun Su., Changyun Miao, Xianguo Li, Xiuzhuang Mei; *Evaluation of image based Abbott–Firestone curve parameters using machine vision for the characterization of cylinder liner surface topography*, Measurement, Volume 55, September 2013, hal. 318-334 Taiwan.
- [3] Gradenwitz, Alfred.: *Applied Machine Vision* #6 (March-April 1966) Beulgrad University, Bulgaria
- [4] M. Sezgin and B. Sankur, *Survey over image thresholding techniques and quantitative performance evaluation*, Journal of Electronic Imaging 13 (2004): hal 146–165.
- [5] Sidla O, Brandle N, Beensova W, Rosner M, Lypetsky Y. *Embedded Vision Challenges*. In “Smart Cameras” Belbachir AN, editor. New York: Springer; 2010. hal. 99-117.
- [6] Batlle J, Marti J, Riadao P, Amat J. *A New FPGA/DSP-based Parallel Architecture for Real-Time Image Processing. Real-Time Imaging*. 2002 February; 8(1):hal. 345-356.
- [7] Ubbens TW, Schuurman DC. *Vision-Based Obstacle Detection using a Support Vector Machine*. In IEEE Canadian Conference on Electrical and Computer Engineering. CCECE '09.; 2009; St. John's, NL. hal. 459-462.
- [8] J. E. Bresenham, *Algorithm for computer control of a digital plotter*, IBM Systems Journal 4 1965: hal 25-30.
- [9] J. Canny, *A computational approach to edge detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence 8; 1986:hal 679-714.
- [10] A.u. R. Shaik, *Image, Processing Technique to Count the Number of Logs in a Timber Truck*, Dalarna University, Borlänge, Sweden
- [11] B. Gary and K. Adrian, *Learning OpenCV*, O'REILLY\* 2008 hal.10-300.
- [12] Kemenhut.go.id, *Perhitungan Nilai Kubisasi Kayu dan*, diakses 15 Maret 2015
- [13] www.modmypi.com, *HC-SR04 Ultrasonic Range Sensor on the Raspberry Pi*, diakses tanggal 10 Juni 2015
- [14] M. E. Criswell, dan M. D. Vanderbilt, 1983 *Properties and Test of Engineering Materials*, 1983 Colorado State University, Colorado

*Halaman ini sengaja dikosongkan*

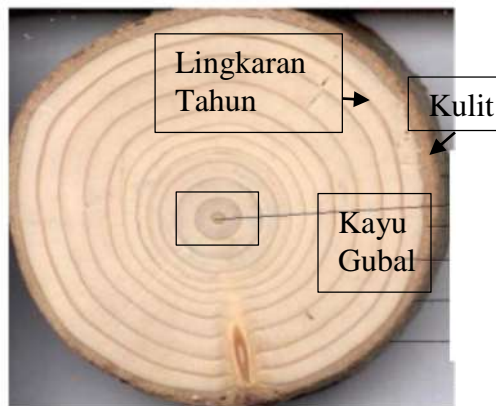
## LAMPIRAN

### 1. Tampilan sistem



Gambar 0.1 Tampilan penampang sistem

### 2. Tampilan penampang kayu



Gambar 0.2 Tampilan penampang kayu [14]

## 2. Datasheet Raspberry Pi 2



# Raspberry Pi



### Raspberry Pi 2, Model B

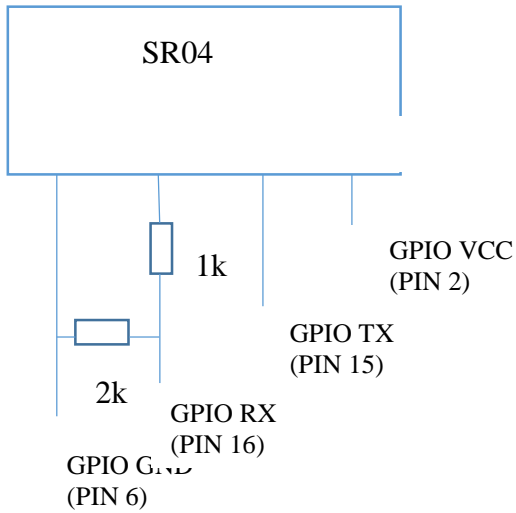
**Product Name** Raspberry Pi 2, Model B

**Product Description** The Raspberry Pi 2 delivers a three-fold increase in processing capacity over previous models. The second generation Raspberry Pi has an upgraded Broadcom BCM2836 processor, which is a powerful ARM Cortex-A7 based quad-core processor that runs at 900MHz. The board also features an increase in memory capacity to 1GB.

#### Specifications

<b>Chip</b>	Broadcom BCM2836 SoC
<b>Core architecture</b>	Quad-core ARM Cortex-A7
<b>CPU</b>	900 MHz
<b>GPU</b>	Dual Core VideoCore IVB Multimedia Co-Processor Provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure
<b>Memory</b>	1GB LPDDR2
<b>Operating System</b>	Boots from Micro SD card, running a version of the Linux operating system
<b>Dimensions</b>	85 x 56 x 17mm
<b>Power</b>	Micro USB socket 5V, 2A
<b>Connectors:</b>	
<b>Ethernet</b>	10/100 BaseT Ethernet socket
<b>Video Output</b>	HDMI (rev 1.3 & 1.4)
<b>Audio Output</b>	3.5mm jack, HDMI
<b>USB</b>	4 x USB 2.0 Connector
<b>GPIO Connector</b>	40-pin 2.54 mm (100 mil) expansion header: 2x20 strip Providing 27 GPIO pins as well as +3.3 V, +5 V and GND supply lines
<b>Camera Connector</b>	15-pin MIPI Camera Serial Interface (CSI-2)
<b>JTAG</b>	Not populated
<b>Display Connector</b>	Display Serial Interface (DSI) 15 way flat flex cable connector with two data lanes and a clock lane
<b>Memory Card Slot</b>	Micro SDIO

### 3. Rangkaian Integrasi ultrasonik dan raspberry pi 2



Gambar 0.3 Rangkaian ultrasonik

*Gambar tidak untuk dipublikasikan*

Gambar 0.4 Desain alat dengan dinamisasi jarak menggunakan ultrasonik HC-SR04




Raspberry Pi 2 Model B (JE Header)									
GPIO	Function	Pin	Pin	Pin	Pin	Pin	Pin	Pin	Pin
1	3.3VDC Power	1	2	3	4	5	6	7	8
8	GPIO 8 (MOSI)	9	10	11	12	13	14	15	16
9	GPIO 9 (MISO)	17	18	19	20	21	22	23	24
7	GPIO 7 (SCL)	25	26	27	28	29	30	31	32
	Ground	33	34	35	36	37	38	39	40
6	GPIO 6	41	42	43	44	45	46	47	48
2	GPIO 2	49	50	51	52	53	54	55	56
3	GPIO 3	57	58	59	60	61	62	63	64
	3.3VDC Power	65	66	67	68	69	70	71	72
12	GPIO 12 (SD0)	73	74	75	76	77	78	79	80
13	GPIO 13 (SD1)	81	82	83	84	85	86	87	88
14	GPIO 14 (SD2)	89	90	91	92	93	94	95	96
	Ground	97	98	99	100	101	102	103	104
	GPIO 15 (SD3)	105	106	107	108	109	110	111	112
21	GPIO 21 (SPI0)	113	114	115	116	117	118	119	120
22	GPIO 22 (SPI1)	121	122	123	124	125	126	127	128
23	GPIO 23 (SPI2)	129	130	131	132	133	134	135	136
24	GPIO 24 (SPI3)	137	138	139	140	141	142	143	144
25	GPIO 25 (SPI4)	145	146	147	148	149	150	151	152
	Ground	153	154	155	156	157	158	159	160

Gambar 0.5 Pin GPIO Raspberry Pi 2


*Gambar tidak untuk dipublikasikan*

Gambar 0.6 Pengujian diameter terkecil terhadap kayu grade 15 pertama



*Gambar tidak untuk dipublikasikan*

Gambar 0.7 Pengujian diameter terkecil terhadap kayu grade 15 kedua



*Gambar tidak untuk dipublikasikan*

Gambar 0.8 Pengujian diameter terkecil terhadap kayu grade 15 ketiga

## BIODATA PENULIS



**Rifyal Rachmat**, lahir di Tuban, 30 Maret 1993. Penulis memulai jenjang pendidikan di TK Dharma Wanita, kemudian melanjutkan ke jenjang sekolah dasar di SDN Ditotrunan 01 Lumajang. Setelah lulus SD tahun 2003 penulis melanjutkan ke SMPN 01 Lumajang, dan pada tahun 2006, penulis melanjutkan ke SMAN 02 Lumajang. Setelah lulus SMA pada tahun 2009 penulis melanjutkan studi di Jurusan Teknik Elektro-Fakultas Teknologi Industri, Institut Teknologi

Sepuluh Nopember, Surabaya dengan bidang studi teknik Elektronika. Penulis dapat dihubungi melalui alamat email: [rifyal.ee.its@gmail.com](mailto:rifyal.ee.its@gmail.com).

*Halaman ini sengaja dikosongkan*